



Blender wiki PDF Manual conversion by Marco Ardito

Details, info, download: <http://amrc.altervista.org>

Updated: 10/11/2014 from:

<http://wiki.blender.org/index.php/Doc:DE/2.6/Manual>



Table of Content

Table of Content	2
Was ist Blender?	21
Habe Sie keine Angst vor Blender!	21
Video: Von Blender 1.60 zu 2.50	23
Version / Revision Meilensteine	23
Der Start!	23
Blender 2.0	24
Blender wird Open Source	24
Ein großer Sprung	24
Blender 2.5 - Die Neuentwicklung!	24
2.63 - Bmesh - Blender mit N-gonen	25
2.64 - Das Open Source VFX Release	25
2.65 - Kontinuierliche Verbesserungen	25
2.66 - Dynamic Topology, Rigid Body Simulation	25
IRC chat channels	27
Wer benutzt Blender?	27
Publikum	29
CG und Blender erlernen	29
Interface (Schnittstelle)	31
Das neue Benutzer-Interface	31
Mehrere Bildschirme	31
Individualisierbarkeit	31
Animations System	32
Alles ist animierbar!	32
Begleittext- und Grapheneditor	32
Neue Funktionen	32
Suchwerkzeug	32
Dateibrowser Verbesserungen	33
Python API	33
Diese Seite als Video!	33
Download und Installation der binären Distribution	34
Hardware Unterstützung	34
Entwickler Plattformen	34
Übersetzen der Quellen	36
Übersetzen der Plugins	36
Download	39
Version	39
Installation	39
Willkommenbildschirm	39
Lizenzvertrag	39
Installationsoptionen	39
Programmooptionen	39
Verzeichnis	40
Installieren	40
Installiert	40
Tragebare Installation	41
Download	42
Version	42
Distributionsversionen	42
Installation	42
In /opt installieren oder unter /usr/local	42
Konfiguration	42
Alt+Mouse Konflikt	42
Blender zu den Anwendungen hinzufügen	43
Blender zum Dock hinzufügen	44
FreeBSD	45
Andere nicht unterstützte Betriebssysteme	45
MorphOS	45
OSX	46
Windows	46
Unix (Linux/BSD/Solaris)	46

Pfad Layout	46
Anmerkungen	47
Pfad für Benutzerskripten	47
Umgebungsvariablen	47
Skriptenpfade und fehlende Schaltflächen	47
Konfiguration & Dateipfade	47
OSX	47
Windows	48
Unix (Linux/BSD/Solaris)	48
Pfad Layout	48
Anmerkungen	49
Pfad für Benutzerskripten	49
Umgebungsvariablen	49
Skriptenpfade und fehlende Schaltflächen	49
OSX	50
Windows	50
Unix (Linux/BSD/Solaris)	50
Pfad Layout	50
Anmerkungen	51
Pfad für Benutzerskripten	51
Umgebungsvariablen	51
Skriptenpfade und fehlende Schaltflächen	51
Speichern Sie ihre Arbeit regelmäßig	52
Konzepte der Benutzeroberfläche	52
Die 3 Regeln	53
Leistungsfähige Benutzeroberfläche	53
Überblick	53
Veranschaulichungen in diesem Handbuch	55
Allgemeine Nutzung	55
Maustasten-Emulation	55
NumPad Emulation	55
Die Standardkonfiguration	56
Das Informationsfenster und Hauptmenü	56
3D Ansicht (3D Window)	57
3D Fenster Kopf (3D Window Header)	57
Einstellungen Fenster (Properties Window)	57
Inhaltsübersicht (Outliner Window)	58
Zeitleiste (Timeline Window)	58
Ein Fenster maximieren	59
Unterteilen eines Fensters	59
Zwei Frames vereinen	59
Verändern der Fenstergröße	59
Inhalte tauschen	59
Neue Fenster öffnen	59
Einen Header verstecken	61
Einen Header sichtbar machen	61
Header-Position	61
Fenstertyp-Schaltfläche	61
Menüs und Schaltflächen	61
Windows XP/Vista/7	63
Linux	63
MacOS	64
Konsolenfenster und Fehlermeldungen	64
Gewöhnliche Meldungen	64
Sie auch	66
Konfiguration der Layouts	67
Einen neuen Screen hinzufügen	67
Löschen eines Layouts	67
Fenster neu anordnen	67
Layouts aus Dateien laden	68
Zusätzliche Layouts	68
Konfiguration der Szenerie	69
Hinzufügen einer neuen Szene	69
Ein kurzes Beispiel	70
Eine Szene löschen	70
Operationsknöpfe	75

Schaltbare Knöpfe	75
Auswahlknöpfe	75
Nummernknöpfe	75
Ausdrücke	75
Einheiten	75
Menüknöpfe	76
Farbwähler Steuerelement	77
Pipette	77
Verschachtelte Knöpfe	77
Was ist Rendering?	79
Rendering eines Bildes mittels Komplettem Rendern - Blender Intern	79
Rendering eines Bildes mittels OpenGL Render	79
Anpassung der Auflösung	79
Ausgabeformat und Ausgabedatei	80
Speichern Ihres Bildes	80
Rendering einer Animation mittels Komplettem Render - Blender Intern	80
Rendering einer Animation mittels OpenGL Render	81
Nur renderbare Objekte anzeigen	81
Die Gründe für OpenGL Rendering	82
Optionen für Aktionen (Benutzerebene)	84
Rückgängig	84
Wiederherstellen	84
Letztes wiederholen	85
Verlauf rückgängig machen	85
Letztes wiederholen	86
Letztes wiederholen	86
Blender Suche	86
Optionen für Dateien (Systemebene)	86
Speichern und Automatische Sicherung	86
Automatische Sicherungen wiederherstellen	87
Andere Optionen	88
Vom Menü aus	89
Zurücksetzen der Standard Szene auf Werkseinstellungen	89
Blender-Bildschirmfotos (Screenshots)	90
Blender-Bildschirmaufnahme (Screencasts)	90
Blender-Fensterdimensionen	92
Addon: 3D View:Screencast Keys (Tastenanzeige)	92
Aufnahme von Bildschirmfotos mittels Betriebssystem	93
Windows-Bildschirmfotos	93
Mac OSX-Bildschirmfotos	93
GNU/Linux-Bildschirmfotos	93
Bildschirmfotos mittels Software-Tools	93
GIMP-Bildschirmfotos	93
Allgemeine Webbasierte Hilfe Optionen	94
Programmier Optionen	94
Optionen zum Diagnostizieren	95
Legacy Version Support	95
Splash Screen	96
Other Help Options	96
Blender Search	96
Tooltips	97
Configure	98
Save the new preferences	98
Load Factory Settings	98
Properties Window	99
View manipulation	99
2D Ansichtspunkte	99
Manipulator	99
Menüs	99
New objects	101
Undo	101
Grease Pencil	101
Playback	101
Keyframing	101
Transform	102
Duplicate Data	102
Hinzufügen und Entfernen von Voreinstellungen	103
Voreinstellungen auswählen	103

Voreinstellung als Standard festlegen	103
Tastenkombinationen Export/Import	103
Maus	103
Numpad Emulation	103
Manipulation der Ansicht	103
Tastenkombinationen Editor	104
Customizing themes with Python	106
Datei Pfade	113
Speichern & Laden	113
Automatische Sicherung	113
General	115
Sound	115
Screencast	115
Open GL	115
Sequencer	116
Solid OpenGL lights	116
Miscellaneous	116
Teil I: Ein statischer Lebkuchenmann	117
Aufwärmübungen	117
Body-Building	118
Gespiegeltes Modellieren mit dem Mirror Modifier	120
Arme und Beine	121
Rückgängig/Wiederherstellen	121
Übereinanderliegende Vertices	121
Der Kopf	122
Arbeiten mit unterteilten Flächen im Subsurf Modifier	123
Achsentreues Skalieren	124
Zwischenprüfung für Gus	125
Ausrichten der Kamera	125
Die Grundfläche	126
Ausleuchtung	126
Erstes Rendern	127
Wir speichern unsere Arbeit	128
Materialien und Texturen	129
Augen und weitere Details	132
Spiegeln einer Kopie an der Position des Drehpunkts (Pivot Point)	133
Mund	133
Material für die Augen	134
Abschließendes Rendern	134
Speichern	135
Teil II: Animieren des Lebkuchenmanns	136
Rigging	136
Skinning	138
Vertex-Gruppen	139
Posieren	141
Originalposition	141
Inverse Kinematics	141
Forward Kinematics	141
Gus walks!	144
Beschreibung	146
Benutzung des Dateimanagers und Ordernavigation	146
Seitenpanel	146
System	146
Bookmarks	146
Recent	146
Optionen beim Öffnen	146
Das Header Panel	147
andere Optionen beim Öffnen	147
Sicherheit	147
Sicherheit	147
Beschreibung	149
Speichern	149
Optionen beim Speichern	149
Compress	149
Remap Relative	149
Save Copy	149
Mehrere Versionen speichern	149
Collada (.dae)	150
Motion Capture (.bvh)	150

skalierbare Vektorgrafik	150
Stanford (.ply)	150
Stl (.stl)	150
3d studio (.3ds)	150
Autodesk FBX (.fbx)	150
Wavefront (.obj)	150
X3D Extensible 3D (.x3d)	150
Der Collada Exporter	151
Operator Vorlagen	151
Optionen Daten-Export	151
Apply Modifiers	151
Selection Only	151
Kinder einschließen	151
Skelett einschließen	152
Formschlüssel einfügen	152
Textur Optionen	152
Nur aktive UV Ebene	152
Einfügen von Texturen	152
Kopieren	152
Skelett Optionen	152
Nur Knochen deformieren	152
Für Second Life exportieren	152
Collada Optionen	152
Triangulieren	152
Objektdaten	152
Transformationstyp	153
Nach Objektnamen sortieren	153
The Collada Importer	153
Technische Details	153
Mesh	153
Import	153
Export	153
Licht	153
Import	153
Export	153
Material & Effect	153
Export	153
Animation	153
Export&Import	153
Knoten	154
Wichtige Dinge zum Beachten	154
Collada (.dae)	155
Motion Capture (.bvh)	155
Stanford (.ply)	155
Stl (.stl)	155
3d studio (.3ds)	155
Autodesk FBX (.fbx)	155
Wavefront (.obj)	155
X3D Extensible 3D (.x3d)	155
Bild Formate	156
Grafiken mit High Dynamic Range	156
Andere Formate	156
Film Formate	156
Farb Modi	156
Farbtiefen	156
Verweiß	156
Render Ausgabe	157
Mehrfachebenen, Multi-pass, Kacheln-basierte Dateien	157
Radiance HDR	158
Fake User	159
Wiederverwendung durch Teilen	159
Dateien lokal speichern	159
lokal gespeicherte Daten entpacken	159
Kopieren und Verknüpfen von Datenblöcken	160
Kopieren und Verknüpfen von Objekt Datenblöcken	161
Kopieren und Verknüpfen von anderen Datenblöcken	161
Abgetrennte Datablöcke	161
Beschreibung	163
Die Outliner Ansicht	163
Auswählen des Outliner Fenstertyps	163

Bedienung des Outliners	164
Auswahl und Abwahl von Datenblöcken	164
Wechsel von Objekt-Level Beschränkungen	166
Suchen	167
Filtern der Ansicht	167
Beispiel	168
Perspektivische und Orthographische Ansichten	170
Beschreibung	170
Optionen	170
Technische Details	171
Definition der Perspektive	171
Definition der Orthographie	171
Rotieren der Ansicht	171
Beschreibung	171
Optionen	171
TrackBall/Tumtable	172
Schwenken der Ansicht	172
Beschreibung	172
Vergrößern der Ansicht	172
Beschreibung	172
Rahmen vergrößern	172
Ausrichten der Ansicht	173
Ansicht ausrichten	173
Lokale und Globale Ansicht	173
Vierer-Ansicht	173
Clipping-Rahmen ansehen	174
Beschreibung	174
Beispiel	174
perspektivische und orthografische Ansicht	175
Beschreibung	175
Optionen	175
Technische Details	176
Perspektivische Projektion	176
Orthografische Projektion	176
Die Blickrichtung drehen	176
Beschreibung	176
Optionen	176
TrackBall/Tumtable	177
Die Ansicht verschieben	177
Beschreibung	177
Die Ansicht Zoomen	177
Beschreibung	177
Auswahlzoom	178
Die Ansicht ausrichten	178
Lokale und Globale Ansicht	178
Quad Ansicht	178
Schneide Grenze anzeigen	178
Beschreibung	179
Beispiel	179
Beschreibung	180
Ansichtseigenschaften Panel	180
Beschreibung	180
Ansicht	180
Item	181
Darstellung	181
Hintergrund Bild	182
Einstellungen	182
Render Rand	184
Arbeiten mit Ebenen	185
Ebenen ansehen	185
Mehrere Ebenen	185
Ein Objekt zwischen Ebenen bewegen	185
Animations Ebenen	186
Beispiel Ebenen Organisation	186

Beschreibung	187
Beispiele	187
Beschreibung	190
Optionen	190
Einführung	191
Rotation mittels Tastaturkürzel	191
Sperren von Rotationsachsen	191
Feinabstimmung einer Rotation	191
Beschreibung	192
Optionen	192
Beschreibung	193
Benutzung	193
Beschreibung	194
Benutzung	194
Beschreibung	196
Benutzung	196
Beschreibung	198
Benutzung	198
Cursor- und Sichtabhängigkeit	198
Logo Beispiel	198
Beschreibung	200
Benutzung	200
Transformations Ausmaß	201
Transformations Orientierung	201
Transformationszentrum	201
Transformation "Springen"	201
Beschreibung	202
mit Transformationemanipulatoren	202
Strg halten	202
Bewegungstransformationen	202
Rotations Transformationen	202
Skalierungs Transformationen	202
⇧ Shift halten	202
STRG und SHIFT halten	202
Bewegungstransformationen	202
Rotations Transformationen	203
Skalierungstransformationen	203
Eingabe über die Werkzeugbalken	204
Panel für transformatorische Eigenschaften	205
Optionen im Objekt Modus	205
transformatorische Eigenschaften sperren	205
eingestellte Transformationen anwenden	205
Transformationen zurücksetzen	206
Beschreibung	207
Optionen zum Entfernen	207
Anwenden von Objekt Transformationen	207
Optionen zum Anwenden	207
Manipulatorkontrollen	209
Manipulator Benutzereinstellungen	209
Auswahl des Vergleichskkoordinatensystems	209
Unser Demo Würfel	211
Orientierungen	211
Global	212
Lokal	212
Normal	212
Aufhängung	213
Ansicht	213
Benutzerdefinierte Orientierungen	213
Beschreibung	215
Benutzung	215
Achsenbegrenzungstypen	215
Achsenbegrenzung	215
Ebenenfestlegung	215
Achsenbegrenzungs Modi	215
Pivot Punkt Auswahl	217
Im Objekt Modus	218
Im Edit Modus	218
Einfache Auswahl	218
Mehrfache Auswahl	219

Edit Mode	220
Im Objekt Modus	222
Im Edit Modus	222
Objekt Modus	223
Im Bearbeitungsmodus	223
Snap	224
Transformationseintrasten	224
Einrastmodi	224
Einrastelement	224
Einrastziel	224
Zusätzliche Einrastoptionen	225
Einrasten	226
Einrastung von Transformierung	226
Modi zum Einrasten	226
Einrasten von Elementen	226
Einrastziel	227
Additional snap options	227
Objekt Modus	228
Edit Modus	228
Einfluss	228
Options	229
Beispiele	229
Drawing	232
Sketching Sessions	232
Shared Grease Pencil Settings	232
Drawing Settings	232
Sensitivity When Drawing	233
Additional Notes For Tablet Users	233
Allgemeine Optionen	234
Timing	234
Beispiel	235
Objekttypen	237
Objektursprungspunkt (Origin oder Pivot)	237
Objektursprungspunkt verschieben	237
Objekte entfernen (Delete)	238
Objekte verbinden (Join)	238
Auswahl und das aktive Objekt	239
Auswahl mit einem Klick	239
Auswahl mit dem Rechteckrahmen	239
Beschreibung	239
Beispiel	239
Auswahl mit dem Lasso	240
Beschreibung	240
Verwendung	240
Auswahl mit dem Kreis	240
Beschreibung	240
Auswahl über Menü	241
Auswahl gruppierter Objekte	241
Beschreibung	241
Optionen	241
Auswahl über verknüpften Inhalt	242
Beschreibung	242
Optionen	242
Auswahl nach Typ	242
Beschreibung	242
Optionen	243
Auswahl nach Layer	243
Beschreibung	243
Optionen	243
Weitere Auswahl-Menüfunktionen	243
Objektmodus	245
Das Erstellen und das Löschen	245
Hinzufügen (Add)	245
Vervielfältigung (Duplicate)	245
Verbinden (join)	246
Löschen (delete)	246
Transformations-Werkzeuge	246

Verschieben (Translation)	246
Drehung (Rotation)	246
Größenänderung (Skalierung)	246
Objekthierarchien (Parenting)	247
Vertex Unterordnung (Parent)	247
Optionen	247
Untergeordnete Objekte Bewegen	247
Beziehung/Unterordnung entfernen	247
Hinweis	248
Objekte Gruppieren	248
Beschreibung	249
Beispiele	249
Verknüpftes Duplizieren	249
Beschreibung	249
Beispiel	249
Andere Duplikationsmethoden	250
Verknüpfte Bibliotheks Duplikation	250
Hinweise	250
Benutzung	251
Einrichtung	251
Ausrichtung	251
Benutzung	252
Skalierung	252
Benutzung	253
Gruppen Duplikation und dynamisches Verknüpfen	253
Echte Kopien erzeugen	253
Benutzung	254
Beispiel	254
Externe Links	255
Benutzen des Bearbeitungsmodus	256
Visualisierung	256
Werkzeugpanel	257
Eigenschaftenpanel	257
Vertices	258
Kanten	258
Flächen	258
Schleifen	258
Kantenschleifen	258
Flächenschleifen	258
Ebene	260
Würfel	260
Kreis	260
UV Kugel	260
Ikkugel	260
Zylinder	260
Kegel	260
Torus	260
Gitter	261
Affe	261
Erweiterungen	261
Selektionsmodi	262
Selektionsmode-Kopfwidgets	262
Selektionsmodus Pop-up	262
Umschalten des Selektionsmodus	262
Ausgewählte Teile nach Umschalten des Selektionsmodus	263
Basic Selection	263
Hinzufügen zur Auswahl	263
Auswählen von Elementen eines Bereichs	263
Rechteckbereich (Rahmenauswahl)	264
Circular region	264
Lasso region	265
Additional Selection Tools	265
Basic Tools	265
Advanced Tools	265
Select Similar	266

Selecting Loops	267
Edge Loop and Vertex Loop Selection	267
Example	267
Face Loop Selection	267
Examples	267
Edge Ring Selection	268
Example	268
Loop to Region and Region to Loop	268
Example: Loop to Region	268
Example: Region to Loop	269
Auswahlmodi	270
Moduspopup	270
Auswahl bei Moduswechsel	270
Kantenschleifen auswählen	272
Kantenringe	272
Flächenschleifenauswahl	273
Ngone im Flächenauswahlmodus	273
Verschmelzen	274
Verschmelzen von Vertices	274
Automatisches Verschmelzen	274
Doppelte Entfernen	274
Teilungswerkzeuge	274
Zerreißen	274
Beispiele	275
Grenzen von Zerreißen	276
Zerreißen und Füllen	276
Herauslösen	277
Abteilen	277
Vertices Verbinden	277
Vertex Verschieben	278
Glätten	278
Vertexunterordnung	278
Haken hinzufügen	279
Formübergang, Propagate Shapes	279
Spiegelachse	280
Pivotpunkt	280
Laplace Glättung	282
Extrudieren von Flächen	283
individuelle Elemente Extrudieren	283
Extrudiere nur Kanten oder Vertices	283
Beispiel	286
Winkel	286
Dupli	287
Verschmelzen oder Entfernen verdoppelter Vertices	288
Normalen Neuberechnen	288
Optionen	290
Benutzung	292
Optionen	292
Fasen Modifikator	294
Benutzung	294
Optionen	294
Beispiele	295
Sculpt Mode	296
Sculpt-Brushes	296
Sculpten mit dem Multires Modifier	297
Sculpt Properties Panel	297
Brush Menu (Pinsel-Menü)	297
Stroke Menu (Stricharten-Menü)	298
Curve Menu (Kurven-Menü)	298
Texture Menu (Texturen-Menü)	298
Symmetry Menu (Symmetrie-Menü)	299
Options Menu (Optionen-Menü)	299
Appearance Menu (Darstellungsmenü)	299
Tool Menu (Werkzeugmenü)	300
Ausblenden und Hervorholen von Mesh-Teilen	300

Tastenkürzel	300
Wozu Vertexgruppen verwenden?	301
Das Vertexgruppen-Menü	301
Erzeugen und Löschen	302
Hinzufügen und Entfernen von Vertices	302
Vertexgruppen-Management	302
Weiches Shading	304
Teile eines Mesh weich schattieren	304
Auto Weichzeichnen	304
Kanten glätten Modifikator	305
Glätten der Mesh Geometrie	305
Werkzeuge zur Mesh-Bearbeitung	305
Modifikatoren	305
Hinweis	307
Textbearbeitung	307
Text einfügen	308
Sonderzeichen	308
Text zu Textobjekten umwandeln	308
Textauswahl	309
Textformatierung	309
Schriftarten	309
Laden und verändern von Schriftarten	309
Größe und Verschiebung	309
Objekte als Buchstaben oder Schrift	310
Text entlang einer Kurve	310
Unterstreichen	310
Buchstabenstile	311
Groß- und Kleinschreibung	311
Absatz	311
Orientierung	311
In Textrahmen	312
Abstände	312
Abstand	312
Form	312
Auflösung	312
Füllung	312
Texturen	312
Geometrie	313
Grundformen	314
Visualisierung	314
Meta Kugel Optionen	314
Auflösung	315
Grenzwerteinfluss	315
Aktualisierung	315
Meta Strukturen	316
Technische Details	316
Grundlegende Struktur	316
Einstellungen	318
Benutzung und Funktion	318
Modifizieren	321
Generieren	321
Verformen	321
Simulieren	322
Options	323
Benutzung	323
Beschreibung	324
Optionen	324
Tipps	326
Berechnung der Verschiebung	326
Beispiele	326
Mechanisch	326
Fraktal	326
Organisch	327
Tutorials	327
Tutorials	327
Optionen	329
Breite	329
nur Vertices	329
Begrenzungsmethode	330

Beschreibung	331
Optionen	331
Siehe auch	331
Beschreibung	332
Optionen	332
Beschreibung	333
Optionen	333
Beispiele Einfache Fläche	333
Dezimierter Zylinder	333
Beschreibung	334
Beschreibung	335
Optionen	335
Beschreibung	336
Optionen	336
Hinweis	336
MultiAuflösungsmodifikator	338
Optionen	338
Ausführungsreihenfolge	339
Kontrolle	340
gewichtete Kanten	340
Kantenschleifen	340
Kombination	341
Grenzen & Hilfen	341
Leistung	342
Beschreibung	343
Optionen	343
Beschreibung	344
Optionen	344
Beschreibung	345
Optionen	345
Siehe	346
Optionen	347
Hinweise	347
Beispiele/Tutorials	347
Partikel und Lattice	347
Bestimmende Faktoren beim Sehen	349
Globale Einflüsse	349
Beleuchtungseinstellungen	349
Beleuchtung im Arbeitsprozess	349
Ausblenden von Materialien zum Einrichten der Ausleuchtung	349
Objekt Daten	352
Preview (Übersicht)	352
Lamp (Lampen)	352
Beschreibung	353
Typen der Abschwächung	353
Lin/Quad Weighted	353
Linear	353
Quadratic	353
Mixing "Linear" and "Quad"	354
Zeroing both "Linear" and "Quad"	354
Graphical Summary	354
Custom Curve	355
Inverse Square	356
Inverse Linear	356
Constant	357
Sphere	357
Examples	358
Distance	358
Inverse Square	358
Sphere	359
Hints	359
Lichtgruppen	361
Materialien	361
Szene	361
Siehe auch	361
Beispiel	362
Siehe	362
Lamp options	364
Sky & Atmosphere	364
Shadow	364
Optionen	365

Himmel	365
Cycles	368
Blender interner Render	368
Shader	368
Diffuse Shading	369
Specular Shading	369
Reflektivität	369
Transparenz	369
Transluzenz	369
Subsurface Scattering	369
Incandescence	369
Ein Material eines anderen Objektes übernehmen	370
Linking material to object or object's data	371
Materials Options	371
Texturen für Materialien	372
Welt Texturen	373
Pinsel Texturen	373
Erste Schritte	374
Texturvorschau	375
Pinseleinstellungen	375
Paint (Malen)	375
Texture (Textur)	376
Tool (Werkzeug)	377
Paint Stroke (Pinselstrich)	377
Paint Curve (Malkurve)	377
Speichern	377
Benutzen eines externen Bildbearbeitungsprogramms	377
Was sind UV's ?	379
Kartographie Beispiel	379
Halb-Kugel Beispiel	379
Der UV Editor	380
Kopfzeile	380
Eigenschaften Panel	380
Navigieren im UV Raum	381
Ansichtsoptionen	381
Vorteile von UV's	381
Oberflächen und (WIRE) Materialien	382
Diffuse	382
Specular	382
Shading	382
Geometry	382
Blending	382
Volume materials	383
Halo materials	383
Skelette	386
Bearbeitung	386
Verbinden	386
Posen	386
Skelettpanel in der Übersicht	387
Darstellungspanel (alle Modi)	387
Knochengruppenpanel (Pose Modus)	387
Posen Bibliothek (Posen Modus)	387
Geistpanel (alle Modi)	388
"iTaSC" Parameterpanel (alle Modi)	388
Bewegungspfadpanel (Posen Modus)	388
Benutzereigenschaftenpanel (alle Modi)	388
Knochenpanel Übersicht	389
Beziehungspanel (Bearbeitungsmodus)	389
Darstellungspanel (Objektmodus)	389
Verformungspanel (alle Modi)	389
Benutzereigenschaftenpanel (alle Modi)	389
Transformationspanel (Bearbeitungs- und Posenmodus)	390
Transformationsbegrenzungspanel (Posenmodus)	390
Panel für Umgekehrte Kinematik (Posenmodus)	390
Knochen Visualisierung	391
Knocheneigenschaften	392
Knochen Steifigkeit	392
Knocheneinfluss	392
Posen Abteilungsüberblick	395

Siehe	395
Automatische UK	397
UK Beschränkungen	397
Einstellung	398
Tips	398
Beschreibung	399
Benutzung von Beschränkungen	399
Beschränkung von Knochen	399
Verfügbare Beschränkungen	399
Transformationsbeschränkungen	399
Beschränkung durch Folgen	400
Beziehungsbeschränkungen / Andere	400
Der Kopf	401
Beschränkungseinstellungen	401
Das Ziel	401
Der Beschränkungsraum	402
Einfluss	402
Beschreibung	403
Optionen	403
Beschreibung	404
Optionen	404
Beschreibung	405
Optionen	405
Beschreibung	406
Optionen	406
Beschreibung	407
Optionen	407
Beschreibung	408
Optionen	408
Beschreibung	409
Optionen	409
Description	410
Optionen	410
Beschreibung	411
Optionen	411
Siehe auch	411
Beschreibung	412
Optionen	412
Beschreibung	413
Optionen	413
Beschreibung	414
Optionen	414
Siehe auch	414
Optionen	415
Beschreibung	416
Optionen	416
Beschreibung	417
Optionen	417
Beschreibung	418
Optionen	418
Einleitung	419
Chapters	419
Animation Basics	419
Animation Editors	420
Categories	420
See Also	420
Zeitstrahl Elemente	421
Das momentane Bild	421
Schlüsselbilder	421
Marker	421
Anpassen der Ansicht	421
Ansichtsmenü	422
Bearbeiten	422
Frame Menü	422
Wiedergabe	422
Wiedergabe Menü	422
Kopfzeile Kontrollelemente	423
Schlüsselbilder erstellen	425

In der 3D Ansicht	425
In dem Eigenschaften Panel	425
In den Animations Editoren	426
Schlüsselbilder bearbeiten	426
Das Konzept der Interpolation	428
Einstellungen	428
Interpolations Modi	428
Extrapolation	429
Hebel Typen	429
Richtung der Zeit	430
Hinzufügen des Modifikators	431
Types of Modifiers	431
Generator	431
Eingebaute Funktionen	431
Envelope	431
Cycles	431
Noise	432
Python	432
Limits	432
Stepped	432
Gravitation	433
Wachsen lassen	434
Styling	434
Animieren	434
Rendern	434
Optionen	434
Emission	434
Haar Dynamik	434
Darstellung	435
Children	435
Render	435
Benutzung	435
Einführung	436
Einstieg	436
Supervised 2D tracking	436
Manual lens calibration using grease pencil and/or grid	436
Camera motion solving	436
Basic tools for scene orientation and stabilization	436
Basic nodes for compositing scene into real footage	436
Not implemented tools	436
Manual	436
Movie Clip Editor	436
Tools available in tracking mode	438
Markers panel	438
Track panel	438
Solve panel	438
Cleanup Panel	438
Clip Panel	439
Properties available in tracking mode	439
Grease Pencil Panel	439
Objects Panel	439
Track Panel	439
Camera Data Panel	440
Display Panel	440
Tracking Settings Panel	441
Common options	441
KLT tracker options	441
SAD tracker options	441
Marker Panel	441
Proxy/ Timecode Panel	441
Tools available in reconstruction mode	442
Rendering	443
Überblick	443
Verteilte Render Farm	443
Render Workbench Integration	443
The Render Settings Panel	444
Render	444
Layers	444
Dimensions	444
Anti-Aliasing	444
Motion Blur	444
Shading	445
Output	445
Leistung	445
Post Processing	445
Stempel	445

Backen	445
Optionen	446
Filtering	446
Filter Size	447
Examples	447
Das Menschliche Auge	449
Im Film	449
In der Computergrafik	449
Bewegungsunschärfe in Blender	449
Abtastbewegungsunschärfe	449
Vektoren Bewegungsunschärfe	449
Beispiele	449
Hinweise	450
Erste Schritte	453
Erste Schritte	455
Tutorials	455
Referenz	455
Perspektivisch	457
Orthografisch	457
Panorama	457
Tiefenunschärfe	457
Clipping	457
Surface Shader	458
Volume Shader	458
Displacement	458
Energieeinsparung	458
Typ	459
Unterteilen	459
Begriffe	460
BSDF Parameter	460
Surface Shader	462
Volume Shader	462
Ambient Occlusion	462
Tricks	462
Point Lamp	463
Area Lamp	463
Sun Lamp	463
Shaders	464
Texturen	464
Mehr	464
Open Shading Language	464
BSDF	465
Diffuse	465
Translucent	465
Glossy	465
Transparent	466
Glass	466
Velvet	467
Emission	467
Background	467
Holdout	468
Mix and Add	468
Path Tracing	469
Der Ursprung des Rauschens	469
Reflektionen	469
Kaustiken und Glossy Filter	470
Lichtabfall	470
Lampe als Beispiel	470
Glas und Transparente Schatten	471
Fenster Lichter	471
Eingeklemmte Glühwürmchen	471
Einführung	473
Einrichtung	473
CUDA	473
Ältere Karten	473
Fehlende Funktionen mit Shader Model 1.x	473
OpenCL	473
Oft gestellte Fragen (FAQ)	473
Warum reagiert Blender nicht während des Render Vorgangs?	473

Warum rendert eine Szene auf der CPU, aber nicht auf der GPU?	473
Kann ich mehrere GPUs für den Render Vorgang nutzen?	474
Würden mehrere GPUs den verfügbaren Speicher erhöhen?	474
Was rendert schneller, NVidia oder AMD, CUDA oder OpenCL?	474
Fehlermeldungen	474
unsupported GNU version! gcc 4.5 and up are not supported!	474
CUDA Error: Invalid kernel image	474
CUDA Error: Out of memory	474
The NVIDIA OpenGL driver lost connection with the display driver	474
Das Gesamtbild	476
Bekannte Grenzen und Probleme	477
Auswahl nach Sichtbarkeit	479
Selection by Edge Types	480
Edge Marks	481
Selection by Face Marks	482
Selection by Group	483
Selection by Image Border	483
Writing Style Modules	485
Selection	486
Chaining	486
Splitting	486
Sorting	487
Stroke creation	487
User control on the pipeline definition	487
Node Konzepte	488
Nodes	488
Nudeln	488
Node Gruppen	488
Benutzen von Nodes	488
Beispiele	489
Ansichtsoptionen	491
Kollabieren von Nodes	491
Größenveränderung der Node	491
Bearbeitungseinstellungen	491
Sockel	491
Sockelfarben	492
Eingabesockel	492
Ausgabesockel	492
Werte	492
Kurven	492
RGB Kurven	492
Bearbeitung der Ansicht	492
Spezial Werkzeuge	492
Nodegruppen	494
Bearbeiten von Nodegruppen	494
Gruppierungen auflösen	494
Render Ebenen Node	496
Benutzung des Alphasockels	496
Optionale Sockel	497
Benutzung des Z Wert Sockels	497
Benutzung des Geschwindigkeitssockels	498
Bildnode	498
Bild Kanäle	498
Speichern und Laden von Renderpässen	498
Bildgröße	499
Animationen	499
Erzeugte Bilder	499
Sonstiges	500
Verbinden von Videosequenzen mit dem Nodeeditor	500
Textur Node	500
Wert Node	500
RGB Node	501
Zeit Node	501
Anzeigenodes	502
Kompositionsnode	502
Datei Ausgabe Node	502
Level Node	503
RGB Kurven Node	504
Mischen Node	504
Farbton Sättigungs Node	505
Gamma	505
Umkehrung	505
Optionen	505
AlphaÜber Node	505
Z-Kombinations Node	506

Farbbalance	506
Farbkorrektur(HSV)	507
Farbtonkarte	507
Einleitung	509
Allgemeine Informationen	509
Anfang - Wiki Tutorials	509
Anfang - Externe links	509
Eigenschaften der Game Engine	513
Nutzung der Game Engine	513
Eigenschaften	514
Zugeordnete Objekte	514
Verbindungen	515
Sensoren	515
Kontroller	515
Aktuatoren	515
Typen von Eigenschaften	516
Using Properties	516
Controller Typen	533
Variablen	534
Operationen	534
Mathematische Operationen	534
Logische Operationen	534
Bedingungen (if)	534
Beispiele	534
Standard-Kamera	538
Kamera Objekt	538
Parent Camera to Object	538
Parent to Vertex	538
Objekt als eine Kamera	538
Kameralinsen Verschiebung	538
Erste Schritte	540
Typen	540
Welt Optionen	540
Beschränkungen	540
Physik Visualisieren	541
Show Framerate and Profile	541
Nach Schlüsselbildern aufnehmen	542
Schlüsselbilder aufräumen	542
In .bullet Datei aufnehmen	543
Mesh Deformationen	543
weichkörper	543
Aktionen	543
Ragdolls	543
Tieferes Wissen	543
Blender und Bullet Quellcode Dateien lesen	543
Editor	545
Engine	545
Unvollständig / Zukünftige Eigenschaften	545

Einleitung



Blender 2.5 mit einer geöffneten Szene aus Big Buck Bunny

Willkommen zu Blender! Diese Dokumentation besteht aus mehreren Teilen: dem Benutzerhandbuch, einer Referenz, Tutorials, Foren und vielen weiteren Internetressourcen. Der erste Teil wird Sie durch das Herunterladen und Installieren führen. Sollten Sie sich entscheiden den Quellcode herunterzuladen, finden sie hier auch die Anleitung Blender aus diesem zu kompilieren.

Blender besitzt eine sehr ungewöhnliche, für die Produktion von 3D-Grafik optimierte Benutzeroberfläche. Das kann bei neuen Benutzern verwirrend wirken, auf lange Sicht jedoch stellen sich die Stärken schnell heraus. Wir empfehlen Ihnen dringend den Abschnitt [Die Benutzeroberfläche](#) zu lesen, um sich zum einen mit der Oberfläche, aber auch mit den Konventionen, welche in dieser Dokumentation verwendet werden, vertraut zu machen.

Was ist Blender?

Blender wurde im Dezember 1993 erdacht und im August 1994 als benutzbare Applikation zur Erstellung einer großen Bandbreite an 2D und 3D Inhalten geboren. Blender liefert ein breites Spektrum an Funktionalität für die Modellierung, Texturierung, Beleuchtung, Animation und Videonachbearbeitung in einem Paket. Durch die Opensource-Architektur ist Blender plattformunabhängig, erweiterbar, hinterlässt einen kleinen Fussabdruck und bietet einen integrierten Arbeitsablauf. Blender ist eines der berühmtesten Quelloffenen 3D-Grafik Programme der Welt.

Zur Zielgruppe von Blender gehören in erster Linie Medienprofis und Künstler. Es können 3D-Visualisierungen, sowie auch Stillleben und Videos in TV- und Kinoqualität erzeugt werden. Durch die eigene 3D-Engine gehören auch Applikationen mit interaktivem 3D-Inhalt zu dieser Palette.

Ursprünglich wurde Blender von der Firma 'Not a Number' (NaN) entwickelt. Fortgeführt wird es nun als 'freie Software' unter der GNU GPL Lizenz von der Blender Foundation in den Niederlanden.

Zwischen 2008 und 2010 wurde Blender vollständig neuentwickelt um sowohl die Funktionalität als auch den Arbeitsablauf und die Oberfläche zu verbessern. Das Ergebnis dieser Arbeit ist Blender in der Version 2.5.

Funktionen:

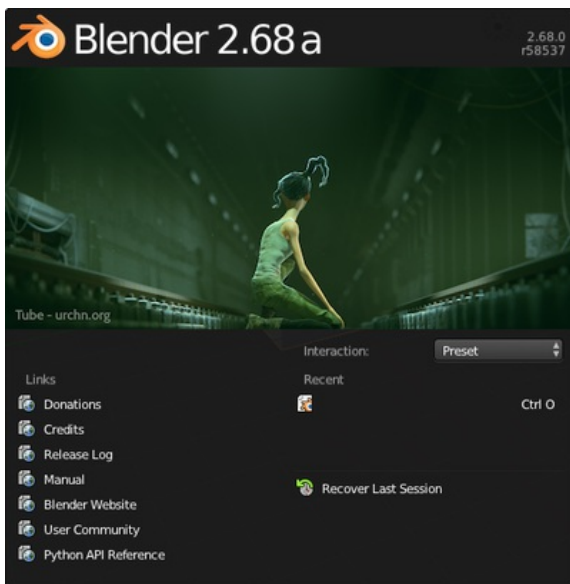


Ein Bild rendern und Postproduktion

- Eine vollständige Entwicklungsumgebung zum Erzeugen von 3D Inhalten, die mit einer breiten Palette notwendiger Werkzeuge ausgestattet ist. Diese decken Themen ab wie Modellierung, UV-Mapping, Arbeiten mit Texturen, Rigging, Skinning, Animation, Partikel- und andere Simulationen, Skripting, Rendern, Compositing, Nachbearbeiten und einen Game Machine Modus zum Erzeugen von Spielen;
- Plattformunabhängig, mit einheitlicher OpenGL GUI für alle Betriebssysteme (anpassbar mit Pythonskripten), fertig verwendbar für alle Windowsversionen (XP, Vista, 7), Linux, OS X, FreeBSD, Sun und zahlreiche weitere Betriebssysteme;
- Hoch qualitative 3D Architektur ermöglicht einen schnellen und effizienten Arbeitsablauf;
- Mehr als 200,000 Downloads weltweit von jeder Version (nach Benutzern);
- Communityunterstützung durch Foren für Fragen, Antworten, und Kritiken bei <http://BlenderArtists.org>, <http://www.BlendPolis.de/> und Nachrichten bei <http://BlenderNation.com>;
- Kleine ausführbare Größe, einfache Verteilung;

Die aktuelle Version können Sie [hier](#) herunterladen.

Habe Sie keine Angst vor Blender!



Blender Willkommensbildschirm ver 2.68 Herbst 2013

Auch wenn das Blender System tausende von verschiedenen Funktionen beinhaltet: Grafikbearbeitung, Modellierung, physikalische Simulation, Videoverarbeitung, Rendern, ... Sie müssen nicht besorgt sein und sagen "Ich kann das nicht!". Das System Blender hat einen gigantischen Vorteil: Eine flache Lernkurve. Es ist ausreichend nur einige Werkzeuge zu verstehen und Sie können schöne Ergebnisse erzielen ohne das Wissen oder die Erfahrung über Detailbereiche (z.B. Kinematik, Knoten, Mixing). Haben Sie den Mut anzufangen!

Die Bedienung des Systems kann für diejenigen, die bereits mit einem ähnlichen System gearbeitet haben, ein wenig kompliziert erscheinen. Es werden alle drei Maus-Buttons genutzt, es sind eine Vielzahl von Tastaturkürzel vorhanden, ... Aber schauen Sie in das Benutzerhandbuch: Alle Funktionen sind aus dem Menü erreichbar und Sie werden nur wenige Tage benötigen um mit "Blender zu spielen" und die grundlegenden Prinzipien "unter der Haube" zu verstehen.

Hier sehen Sie einen netten Willkommensbildschirm nach dem Start von Blender. Das zentrale Bild ist selbstverständlich mit Blender erstellt worden, jede neue Version erhält ein neues Bild. Es kann eine Motivation für Sie sein - starten Sie einfach mit Blender, seien Sie kreativ, seien Sie mutig.

Blenders Geschichte

Im Jahr 1988 wurde das niederländische Animationsstudio NeoGeo von Ton Roosendaal mitgegründet. NeoGeo wurde schnell zum größten 3D-Animationsstudio in den Niederlanden und eines der führenden Häuser in Europa für Animationen. NeoGeo erstellte preisgekrönte Produktionen (European Corporate Video Awards 1993 und 1995) für Großkunden wie den multinationalen Elektronikkonzern Philips. Innerhalb von NeoGeo war Ton verantwortlich für "Art Direction" und interne Softwareentwicklung. Nach sorgfältiger Überlegung entschied Ton, dass das aktuelle 3D-inhouse-Tool für NeoGeo zu alt, zu schwerfällig zu pflegen und zu aktualisieren war. Es musste von Grund auf neu geschrieben werden. 1995 begann man mit der Neuprogrammierung. Daraus wurde die 3D-Software Creation Suite, die wir alle als Blender kennen. Als NeoGeo Blender weiter verfeinerte und verbesserte, stellte Ton fest, dass Blender auch von anderen Künstlern außerhalb von NeoGeo genutzt werden könnte.

Im Jahr 1998 beschloss Ton, eine neue Firma namens "Not a Number" (NaN) zur weiteren Entwicklung und Vermarktung von Blender als Spin-off von NeoGeo zu gründen. Im Mittelpunkt des NaN stand der Wunsch eine kompakte, plattformübergreifende, kostenlose 3D Creation Suite zu erstellen und zu verteilen. Zu seiner Zeit war dies ein revolutionäres Konzept, da die meisten kommerziellen Modellierer Tausende US-Dollars kosteten. NaN hoffte dadurch der allgemeinen Öffentlichkeit ein 3D-Modellierungs- und Animations-Tools mit professionellem Niveau zur Verfügung zu stellen. Ihr Geschäftsmodell bestand aus kommerziellen Produkten und Dienstleistungen rund um Blender. Im Jahr 1999 besuchte NaN ihre erste Siggraph-Konferenz mit dem Bemühen, Blender noch bekannter zu machen. Blenders erste Siggraph Convention 1999 war ein riesiger Erfolg und erhielt große Aufmerksamkeit sowohl von der Presse als auch von Besuchern. Blender war ein Hit und konnte sein enormes Potenzial zeigen!

Beflügelt durch eine erfolgreiche Siggraph im Frühjahr 2000, sicherte sich NaN eine Finanzierung in Höhe von € 4,5 Millionen von Risikokapitalgebern. Diese neuen Finanzzuflüsse ermöglichten NaN seine Operationen rasch erweitern. Bald hatte NaN nicht weniger als fünfzig Mitarbeiter rund um die Welt, die versuchten Blender zu verbessern und zu vermarkten. Im Sommer 2000 wurde "Blender v2.0" veröffentlicht. Diese Version von Blender beinhaltet die Integration einer Game-Engine in die 3D-Suite. Ende 2000 übertraf die Zahl Nutzer auf der NaN-Website 250.000.

Leider entsprachen NaN's Ambitionen und Möglichkeiten nicht der Leistungsfähigkeit des Unternehmens und den Marktrealitäten dieser Zeit. Diese "über-Erweiterung" führte zu einem Neustart NaN's mit neuer Finanzierung als kleineres Unternehmen im April 2001. Sechs Monate später wurde NaN's erstes kommerzielles Softwareprodukt, "Blender Publisher" gestartet. Dieses Produkt wurde auf dem aufstrebenden Markt der interaktiven, Web-basierten 3D-Medien ausgerichtet. Aufgrund der enttäuschenden Verkäufe und der anhaltend schwierigen Wirtschaftslage, entschieden die neuen Investoren, sämtliche Aktivitäten NaN's einzustellen. Das Herunterfahren bedeutete ebenfalls, die Entwicklung von Blender aufzugeben. Obwohl es eindeutig Defizite in der damals aktuellen Version von Blender, mit seiner komplexen internen Softwarearchitektur, unfertigen Features und seiner nicht-standardisierten GUI gab, konnte Ton, die Benutzer-Community und Kunden, die Blender Publisher gekauft hatten, es nicht verantworten, Blender in Vergessenheit geraten zu lassen. Da ein Neustart des Unternehmens mit einem ausreichend großen Team an Entwicklern nicht durchführbar war, gründete Ton Roosendaal im März 2002 die Non-Profit-Organisation *Blender Foundation*.

Das primäre Ziel der Blender Foundation war es, einen Weg zu finden, um Blender als ein community-basiertes [Open Source](#) Projekt weiter zu entwickeln und zu vermarkten. Im Juli 2002 gelang es Ton, die NaN Investoren vom einzigartigen Blender Foundation Plan zu überzeugen, Blender als Open Source freizugeben. Die "Free Blender"-Kampagne versuchte 100.000 € zu sammeln, damit die Stiftung die Rechte an dem Blender Quellcode und das Recht des geistigen Eigentums von den NaN Investoren kaufen konnte. Anschließend wurde Blender der Open Source Gemeinschaft veröffentlicht. Mit einer enthusiastischen Gruppe von Freiwilligen, darunter mehreren Ex-Mitarbeitern von NaN, wurde eine Kampagne ins Leben gerufen, um "Free Blender" zu etablieren. Zu jedermanns Überraschung erreichte die Kampagne das "100.000 €-Ziel" in nur sieben Wochen. Am Sonntag, den 13. Oktober 2002, wurde Blender weltweit unter den Bedingungen der [GNU General Public License \(GPL\)](#) veröffentlicht. Bis heute ist die Blender Entwicklung angetrieben von einem Team aus weltweit verstreuten, engagierten Freiwilligen unter der Führung des ursprünglichen Schöpfers Ton Roosendaal.

Video: Von Blender 1.60 zu 2.50

[\[video link\]](#)

Version / Revision Meilensteine



Release Notes

Zum Anschauen der Release Notes zu jeder Version, können Sie auf die Versionsnummer klicken.

- Von 1.00 bis 2.30 gibt es keine Links zu den Release Notes mehr.
- Von 2.30 bis 2.40 können Sie die Release Notes nur unter [Blender Release Notes](#) finden.
- Von Version 2.40 aufwärts befinden sich die Release Notes im Entwicklerbereich im Wiki und unter [Blender Release Notes](#).

Blenders Geschichte und Roadmap:

Der Start!

- 1.00 - Januar 1995: Blender ist in der Entwicklung beim Animationsstudio NeoGeo.
- 1.23 - Januar 1998: SGI-Version im Internet, IrisGL veröffentlicht.
- 1.30 - April 1998: Linux- und FreeBSD-Version, Port auf OpenGL und X.
- 1.3x - Juni 1998: NaN gegründet.
- 1.4x - September 1998: Sun- und Linux-Alpha-Version veröffentlicht.
- 1.50 - November 1998: Erstes Handbuch veröffentlicht.
- 1.60 - April 1999: C-Key (neue Funktionen hinter einer Sperre, \$ 95), Windows-Version veröffentlicht.

- 1.6x - Juni 1999: BeOS-und PPC-Version veröffentlicht.
- 1.80 - Juni 2000: Ende des C-Key, Blender ist wieder volle Freeware.

Blender 2.0

- 2.00 - August 2000: Interaktive 3D-und Echtzeit-Engine implementiert
- 2.10 - Dezember 2000: Neue Engine, Physik und Python.
- 2.20 - August 2001: Character-Animations-System.
- 2.21 - Oktober 2001: Blender Publisher gestartet
- 2.2x - Dezember 2001: Mac OSX Version.

Blender wird Open Source

- 13 Oktober 2002: Blender wird Open Source, 1. Blender Conference
- 2.25 - Oktober 2002: [Blender Publisher](#) wird frei verfügbar.
- Tuhopuu1 - Oktober 2002: Der experimentelle Blender-bereich wird erstellt, für Programmierer eine Sandbox.
- 2.26 - Februar 2003: Das erste echte Open Source Blender.
- 2.27 - Mai 2003: Das zweite Open Source Blender.
- 2.28x - Juli 2003: Erste Version der 2.28x Serie.
- [2.30](#) - Oktober 2003: Das [Preview Release](#) der 2.3x UI Überarbeitung wird auf der 2. Blender Conference präsentiert.
- [2.31](#) - Dezember 2003: [Upgrade](#) zur Stabilisierung des 2.3x UI-Projektes.
- [2.32](#) - Januar 2004: [Generalsanierung](#) interne Rendering-Fähigkeiten.
- [2.32](#) - April 2004: Die [Game Engine](#) kehrt zurück, Ambient Occlusion, neue prozedurale Texturen.
- [2.34](#) - August 2004: [Große Verbesserungen](#): Partikel-Wechselwirkungen, LSCM UV-Mapping, funktionale Integration YafRay, Faltenwicklung in Subdivision Surfaces, Ramp Shadern, volle OSA, und vieles vieles mehr.
- [2.35](#) - November 2004: [Eine neue Version voller Verbesserungen](#): Objekt Haken, Kurven Deformers und Kurven Tapers, Partikel-Vervielfältiger und vieles mehr.
- [2.36](#) - Dezember 2004: [Eine stabile Version](#), viel Arbeit hinter den Kulissen, Normal-und Displacement Mapping Verbesserungen.

Ein großer Sprung

- [2.37](#) - Juni 2005: [Ein großer Sprung](#): Transformation Tools und Widgets, Softbodies, Kraftfelder, Deflections, inkrementelle Subdivision Surfaces, transparenter Schatten und Multithreading-Rendering.
- [2.40](#) - Dezember 2005: [Ein noch größerer Sprung](#): Vollständige Überarbeitung des Skelettsystems, Form-Tasten, Fell mit Partikeln, Flüssigkeiten und Festkörper.
- [2.41](#) - Januar 2006: [Viele Fehlerbehebungen](#), und einige Game-Engine Features eingefügt.
- [2.42](#) - Juli 2006: [Das "Knoten"-Release](#). Über [50 Entwickler](#) haben Knoten-Array Modifier, Vektor Unschärfe, neue Physik-Engine, Rendering, LipSync, und viele andere Funktionen beigetragen. Dies war das Release zu [Project Orange](#).
- [2.43](#) - Februar 2007: [Das "Multi"-Release](#): Multi-Resolution Mesh, mehrschichtige UV Texturen, Multi Schicht-Bildern und Multi-Pass-Rendering und Baking, Sculpting, Retopology, mehrere zusätzliche Matte-, Verzerren- und Filter-Knoten, Modellierung und Animation Verbesserungen, mehreren Brushes, Fluid-Partikel, Proxy-Objekte, Sequenzer, und Post-Production UV Texturierung. Puh! Oh, und eine neue Website. Und ja, es hat noch Multi-Threaded Rendering für Multi-Core-CPU's. Mit Verse ist es Multi-User, so dass mehrere Künstler auf der selben Bühne gemeinsam arbeiten können. Schließlich noch Render-Farmen für verteiltes Multi-Workstation-Rendering.
- [2.44](#) - Mai 2007: [Das "SSS"-Release](#): Die großen Nachrichten: zusätzlich zu zwei neuen Modifikatoren und dem Wiedererwachen der Unterstützung von 64-Bit Betriebssystemen, wurde der Zusatz von Sub Surface Scattering, die Lichtstreuung unter der Oberfläche von organischen und weiche Gegenstände simuliert.
- [2.45](#) - September 2007: [weiteres Bugfix-Release](#): Größere Bugfixes, einige Performance-Probleme gelöst.
- [2.46](#) - Mai 2008: [Das "Peach"-Release](#) war das Resultat einer gewaltigen Anstrengung von über 70 Entwicklern und bietet Verbesserungen an Kern und Patches für Haare und Fell, ein neues Partikel-System, verbesserte Bild-Browsing, Cloth, welches eine nahtlose und non-intrusive Physik-Cache bietet, Rendering-Verbesserungen in Spiegelungen, AO, und Baking; ein Mesh Modifier für Muskeln etc., bessere Unterstützung für Animationen über Skelett-Tools und Zeichnung, Skinning, Beschränkungen und ein bunter Action-Editor und vieles mehr. Es war das Release zu [Project Peach](#).
- [2.47](#) - August 2008: [Bugfix Release](#).
- [2.48](#) - Oktober 2008: [Das "Apricot"-release](#): Coole GLSL Shader, Licht und GE Verbesserungen, Snap, Himmel-Simulator, shrinkwrap Modifier, Python Verbesserungen.
- [2.49](#) - Juni 2009: [Das "Pre-re-factor"-Release](#) hat signifikante Verbesserungen am Kern- und GE. Zu den Kernerweiterungen gehören knotenbasierte Texturen, Skelett-Skizzierung (namens Etch-a-Ton), boolean Mesh Verbesserungen, JPEG2000-Unterstützung, Bild-Projektion für die direkte Übertragung der Bilder, Modelle, und ein signifikanter Python-Skript-Katalog. GE-Erweiterungen inklusive Video-Texturen, mit denen Filme in-game abgespielt werden können, Upgrades auf die Bullet Physik-Engine, Kuppel (Fischauge) Rendering und weitere API GE Aufrufe wurden zur Verfügung gestellt.

Blender 2.5 - Die Neuentwicklung!

- [2.5x](#) – Von 2009 bis August 2011. Dieses [Serien-Release](#) bestand aus vier Vorversionen (von Alpha0 - November 2009 - bis Beta Juli 2010) und drei stabilen Versionen (von 2.57 - April 2011 - bis 2.59 - August 2011). Es ist eine der wichtigsten Entwicklung (<http://www.blender.org/development/current-projects/blender-25-project/> project]) von Blender mit einer vollständigen Neu-Kodierung der Software mit neuen Funktionen, einem Redesign des internen Fenster-Managers und dem Ereignis-/Werkzeug-/Datenbehandlungssystems, eine neue Python API... Die finale Version von diesem Projekt war Blender 2.59 im August 2011.
- [2.60](#) – Oktober 2011: [Internationalisierung der Benutzeroberfläche, 3D Audio und Video](#). Dieses Release enthält Verbesserung im Animationssystem und in der Game Engine, Vertex Weight Groups Modifiers, 3D Audio und Video, Fehlerbehebungen, und die Internationalisierung der Benutzeroberfläche (Der Garlic-Zweig wurde mit dem trunk zusammengeführt).
- [2.61](#) – Dezember 2011: [Kameraverfolgung \(Camera Tracking\), Meeressimulation, Cycles Render Engine, Dynamic Paint](#). Die neue Cycles Render-Engine wurde der Standardinstallation von Blender hinzugefügt, genauso die Kameraverfolgung (Camera Tracking) für Mischung von Filmmaterial mit 3D, Dynamic Paint für die Modifizierung von Texturen mit Mesh-Kontakt/-

Annäherung, die Meeressimulation ist ein neuer Modifikator um Meer und Schaum zu simulieren (portiert von dem Open Source Houdini Ocean Toolkit), neue Addons, Fehlerbehebungen, und viele Erweiterungen für die Python API wurden hinzugefügt.

- [2.62](#) - Februar 2012: [Carve Booleans](#), [Bewegungsverfolgung \(Motion Tracking\)](#), [Remesh Modifikator](#). Die [Carve](#) Bibliothek wurde hinzugefügt um das Ergebnis bei der Durchführung von Boolean Operationen zu verbessern, Blender unterstützt nun die Bewegungsverfolgung (Motion Tracking) für Objektbewegungen in einer Szene, der Remesh-Modifikator generiert neue Topologien auf Basis eines Eingangs-Mesh, viele Verbesserungen in der Game-Engine, Coolada, Bump Mapping, Dynamcis Paint, UV Werkzeuge, Cycles Render Engine, Matrizen und Vektoren in der Python API wurden verbessert, neue Addons, und Fehler wurden behoben.

2.63 - Bmesh - Blender mit N-gonen

- [2.63](#) - April 2012: [Ein neues Mesh-System mit voller Unterstützung für N-seitige Polygone anstatt nur Dreiecken und Quadraten wurde in Blender hinzugefügt](#), Sculpt Hiding, Cycles Render mit Panoramakamera, "mirror ball Environment" Texturen und "float precision" Texturen, "render layer mask layers", "ambient occlusion" und Viewport Darstellung von Hintergrundbildern, neue Import- und Export-Addons wurden hinzugefügt, und Renderfarm.fi unterstützt nun Cycles. 150 Fehlerbehebungen für Fehler, die in vorherigen Releases existierten.

2.64 - Das Open Source VFX Release

- [2.64](#) - Oktober 2012: [Mask Editor](#), [verbesserter Bewegungsverfolger \(Motion Tracker\)](#), [Opencolor IO](#), Verbesserungen im Cycles Render, Verbesserungen im Sequencer, bessere Mesh-Werkzeuge (Inset und Bevel wurden verbessert), neue zusammenführende Knoten für den Green Screen, Sculpt Masking, Collada Verbesserungen für Game Engines, neue Skin Modifikatoren, neues Backend für zusammenführende Knoten, und viele Fehler wurden behoben.

2.65 - Kontinuierliche Verbesserungen

- [2.65](#) - Dezember 2012: [Feuer und Rauch](#), ["anisotropic shaders" für Cycles](#), Verbesserungen bei den Modifikatoren, das Bevel-Werkzeug beinhaltet jetzt Rundungen, neue Addons, und mehr als 200 Fehler, die in vorherigen Versionen existierten, wurden behoben, das Ergebnis war ein **2.65a** Release!

2.66 - Dynamic Topology, Rigid Body Simulation

- [2.66](#) - Februar 2013: [Dynamic Topology Sculpting](#), [Rigid Body Simulation](#), Verbesserungen der Benutzeroberfläche und Bedienbarkeit (einschließlich Unterstützung für das neue Mac 'Retina Display'), der Cycles Render unterstützt nun Haare, Verbesserungen in der Transparenz von Bildern, das Bevel-Werkzeug unterstützt jetzt individuelles Vertex-Bevelling, ein neuer [Mesh Cache](#)-Modifikator und der neue [UV Warp](#)-Modifikator, ein neuer SPH "particle fluid solver" wurde für die Berechnung von flüssigen Dynamiken hinzugefügt, Verbesserungen in der Game Engine und Collada, Unterstützung für "vertex color bake", effizienteres "ambient occlusion baking" für "multires" Meshes, kantenbasiertes UV-Stitching, mehr Kontrolle über Mapping-Texturen-Pinsel für die Texturen-Zeichnung, Gradient-Werkzeuge für "Weight"-Zeichnungen, und ein Übersetzungsknoten für den "Compositor". Ein neues Addon für die Unterstützung des MilkShape 3D Formats und ein EDL Video Import. Mehr als 250 Fehler, welche in vorherigen Versionen existierten, wurden behoben, das Ergebnis war ein **2.66a** Release!.

Über freie Software und die GPL



Wenn von "freier" Software die Rede ist, denkt man zuerst an "\"frei von Kosten\"". Obwohl das zwar in den meisten Fällen stimmt, wird "freie Software" wie es von der Free Software Foundation, den Begründern des GNU Projekts und den Schöpfern der GNU General Public License, benutzt wird eher gemeint als "freiheitlich zu benutzende Software". Freie Software ist Software die jeder ohne Beschränkungen frei ist zu benutzen, zu kopieren, zu modifizieren und weiter zu verteilen. Im Kontrast dazu steht kommerzielle Software die alle diese Freiheiten nicht erlaubt. Freie Software ist außerdem flexibler, da in der Software auftretende Fehler selbst behoben werden können.

Programme die unter einer GNU General Public License veröffentlicht werden:

- können für jeden Zweck benutzt werden.
- können beliebig modifiziert werden da der Quellcode öffentlich verfügbar ist.
- können beliebig vervielfältigt und weiterverteilt werden.
- können beliebig verbessert und selbst herausgegeben werden.

Allerdings kommen mit diesen Rechten auch einige Pflichten:

- jeder weiterverteilten Kopie muss eine GPL beigefügt sein, damit jeder Nutzer über seine Rechte, die aus der Lizenz folgen, informiert ist.
- der Quellcode muss entweder beigefügt oder frei erhältlich sein.
- wenn das Programm verändert wurde müssen die Veränderungen ebenfalls von der GPL abgedeckt sein bzw. unter einer GPL lizenziert werden und der Quellcode muss frei verfügbar sein. (Das bedeutet das GPL Programme oder Teile davon nicht in kommerzielle Software umgewandelt werden kann)
- Die GPL Lizenz darf nicht erweitert oder verändert werden (Das bedeutet das ein GPL Programm nicht als kommerzielles Programm neu lizenziert werden kann.).

Für mehr Informationen zur GPL wird auf die [Webseite des GNU Projekts](http://www.gnu.org) verwiesen.

Immer hilfsbereit: Die Blender Community

Von Anfang an war Blender frei verfügbar, auch als es noch nicht OpenSource war. Das führte zu der Bildung einer großen, stabilen und aktiven Community rund um Blender seit 1998.

Die Community zeigte ihre Stärke während des kritischen Moments der „Befreiung“ von Blender durch die „Free Blender“-Kampagne im Spätsommer 2002.

Heute ist die Community unterteilt in zwei, in weiten Bereichen überlappende, Lager:



Die Entwicklungs-Community, welche sich um die Seite der [Blender Foundation](http://www.blender.org) gruppiert. Dort ist das Zuhause verschiedener Projekte der Entwickler, des Quellcodes im CVS-Repository, der englischen Dokumentation und öffentlicher Diskussionsforen. Entwickler, die an Blender selbst arbeiten, Python Skripter, Autoren der Dokumentation und alle die für die Entwicklung von Blender arbeiten, können dort angetroffen werden.

Gehe zu <http://www.blender.org>



Die Benutzer-Community, die sich um die unabhängige Seite [BlenderArtists](http://www.BlenderArtists.org) gruppiert. (Anmerkung: Das deutschsprachige, sehr aktive Pendant dazu ist [blendpolis](http://www.blendpolis.org).) Hier zeigen Künstler, Spieledesigner und Blenderfans ihre Arbeiten, die sie mit Blender erstellt haben, um Feedback zu bekommen. Außerdem findet man hier Hilfe, die einem einen tieferen Einblick in Blenders Funktionalität ermöglicht. Auch Tutorials können hier gefunden werden.

Gehe zu <http://www.BlenderArtists.org> bzw. Gehe zu <http://www.blendpolis.org>

Diese Webseiten sind nicht die einzigen Informationsquellen zu Blender. Die weltweit vertretene Community hat eine große Anzahl an unabhängigen Seiten geschaffen, in anderen Sprachen oder mit einem bestimmten Thema als Schwerpunkt. Eine regelmäßig überarbeitete Liste mit anderen Internetseiten zu Blender kann auf den oben genannten Seiten gefunden werden.

IRC chat channels

Für direktes Online-Feedback gibt es drei verschiedene Chats auf irc.freenode.net. Sie können sich dort mit ihrem bevorzugten IRC-Client einloggen :

- [#blenderchat](#) für allgemeine Diskussionen über Blender;
- [#blenderqa](#) für Fragen zur Benutzung von Blender;
- [#gameblender](#) für die Diskussion von Themen rund um die integrierte Game-Engine

Für Entwickler gibt es auch :

- [#blendercoders](#) für Fragen und Diskussionen über die Programmierung von Blender. Dort gibt es auch jeden Sonntag um 16:00 Uhr (Zeitzone der Niederlande) ein Entwicklertreffen;
- [#blenderpython](#) für Diskussionen zum Python-API und zur Entwicklung von Python-Skripten;
- [#blenderwiki](#) ist für Fragen zum WIKI zuständig.

Wer benutzt Blender?

Neue Versionen von Blender werden innerhalb der ersten 10 Tage nach der Veröffentlichung von über einer Million Menschen aus der ganzen Welt heruntergeladen. Diese Zahl beinhaltet alle Plattformen (Windows, Linux und Mac OS) aber nicht die Weiterverteilung des Programms, welche explizit erlaubt und unbeschränkt ist. Schätzungsweise gibt es über zwei Millionen Benutzer. Diese Handbuch wurde für diese breite Gruppe talentierter Leute geschrieben, die Blender benutzen:

- Studenten oder Leute deren Hobby es ist, die Welt der Computergraphic (CG) und 3D-Animation zu erforschen.
- 2D-Künstler die einzelne Bilder / Poster erstellen und Blender für das Post-Processing benutzen.
- 2D-Künstler oder Teams, welche Cartoons oder Kurzanimationen und TV-Werbung erstellen (z.B. "The Magic of Amelia").
- 3D-Künstler die allein oder mit anderen Personen Kurzanimationen erstellen, teilweise in Kombination mit Realfilm-Sequenzen (z.B. "Suburban Plight").
- 3D-Teams welche komplett animierte Filme produzieren (z.B. "[Elephant's Dream](#)", "[Plumiferos](#)", "[Big Buck Bunny](#)" or "[Sintel](#)")
- 3D-Teams welche Realfilme produzieren und CG-Effekte einbinden.

Es gibt also ein großes Spektrum von Benutzern, vom Teenager zum "Blender-Urgestein", und die Community hat eine relativ gleichmäßige Verteilung von Anfängern und Profis und sogar einige kommerzielle Anbieter. Man kann die 2D- und 3D-Teams weiter in unterschiedliche Aufgabenbereiche unterteilen. Blender wird in folgenden genutzt:

- Regisseur - Legt fest was in welcher Szene passieren soll und definiert die Kamerafahrten.
- Modeler - Erstellt die virtuelle Realität. Spezialisiert sich oft in Character-, Requisiten- oder Landschafts-/Hintergrund-Modeler.
- Kameramann, Director of Photography (DP) - Positioniert Kameras und koordiniert deren Bewegung, nimmt Realfilmsequenzen auf und steuert das Rendering.
- Material-Painter - "Bemalt" die Kulissen, Darsteller, und sonst alles was sich bewegt. Wenn sich etwas mal nicht bewegt, wird es trotzdem bemalt.
- Animation und Rigging - lässt Objekte mit Hilfe von Armatures hin und her hüpfen.
- Licht- und Farbspezialist - leuchtet die Szene aus, passt Farben an, fügt Schmutz und Dreck zu Materialien und Texturen hinzu damit sie realistischer wirken.
- Spezial-Talente - Flüssigkeiten, Motion-Capturing, Kleidung, Staub, Schmutz, Feuer, Explosionen… all das lustige Zeug

eben.

- Editor - bekommt die einzelnen Sequenzen vom DP und schneidet sie zu einem spannenden Film zusammen. Dazu gehört auch das Herausschneiden von unwichtigen Szenen.

Über dieses Handbuch

Dieses Handbuch ist ein Mediawiki das in Zusammenarbeit von freiwilligen [Autoren](#) aus der ganzen Welt geschrieben wird. Es wird täglich aktualisiert, wobei die Englische Version die "offizielle" Version und somit am aktuellsten ist. Für das weltweite Publikum wird es aus dem Englischen dann in verschiedene andere Sprachen übersetzt. Das Handbuch ist eigentlich immer ein wenig veraltet, da die über 50 freiwillige Entwickler unermüdlich an Verbesserungen des Programms arbeiten. Trotzdem soll das Ziel sein, hier die bestmögliche Dokumentation für Blender anzubieten.

Um Sie auf die beste Art und Weise zu unterstützen, wurde das Handbuch entsprechend dem der typischen Arbeitsweise eines 3D-Künstlers organisiert. Es gibt verschiedene Zwischenstopps, um die Orientierung in diesem neuen Gebiet und den Umgang mit einem doch sehr komplexen Programm zu erleichtern. Wenn man das Handbuch von vorn bis hinten liest, folgt man einem Pfad wie ihn die meisten Benutzer beim Erlernen von Blender und dem Erzeugen von Animationen gehen:

1. Blender kennenlernen = Einleitung, Navigation in 3D, Szenenmanagement
2. Models = Modelling, Modifier
3. Ausleuchtung (*Lighting*)
4. Shading = Materialien, Texturen, Hintergründe
5. Animation = Grundlagen, Charakteranimation, Effekte, physikalische Simulationen
6. Rendern = Rendern, Compositing, Videobearbeitung
7. Jenseits von Blender = Erweiterungen für Blender

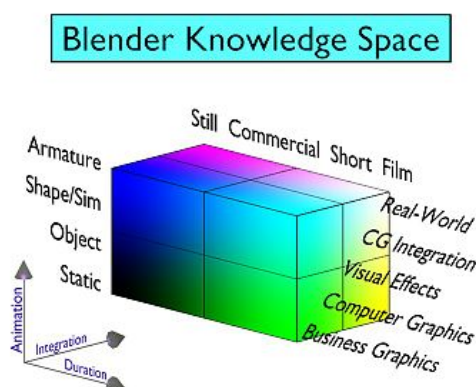
Publikum

Dieses Handbuch ist für ein sehr breites Publikum geschrieben. Es beantwortet Fragen wie "Ich will *irgendwas* machen, wie geht das mit Blender?" bis hin zu "Was wurde zuletzt an den Bearbeitungsfunktionen für *Meshs* verändert?".

Das Handbuch entsteht durch eine weltweite Kollaboration bei der einzelne Leute ihre wertvolle Zeit investieren. Es kann also durchaus zu Verzögerungen zwischen der Implementierung einer neuen Funktionen und deren Dokumentation kommen. Es wird aber versucht, das Handbuch so aktuell wie möglich zu halten. Der Fokus soll bei den Themen liegen, die für den Endbenutzer interessant sind, und nicht zu weit abzuschweifen um etwa den Sinn des Lebens zu diskutieren.

Es gibt [weitere Wiki-Bücher über Blender](#), welche tiefer in bestimmte Themen eindringen oder Blender aus einem anderen Blickwinkel betrachten, z.B. Tutorials, das Referenz-Handbuch, Artikel über die Softwareentwicklung oder zu Skripting-Sprachen. Wenn also eine Frage hier nicht beantwortet wird, suchen Sie bitten auch in den anderen [Blender-Wiki-Büchern](#).

CG und Blender erlernen



Das Erlernen der Erstellung von Computergrafik (CG) und das Kennenlernen von Blender sind zwei unterschiedliche Themen. Zunächst muss man erstmal wissen, was ein Computer-Modell ist, danach kann man es in Blender entwickeln. Zuerst muss man etwas von guter Ausleuchtung verstehen, um dann in Blender mit verschiedenen Lampen gute Ergebnisse zu erzielen. Dieses erste, begriffliche Wissen kann man sich in Kursen über Kunst und Medien, aus Büchern oder durch Ausprobieren aneignen. Auch wenn ein Buch ein anderes Programm (wie Max oder Maya) zur Visualisierung benutzt, kann es trotzdem gut dieses konzeptionelle Wissen lehren.

Sobald man die Grundlagen kennt, kann man sich viel leichter in Blender (oder jedes andere CG-Programm) einarbeiten. Beides gleichzeitig zu lernen ist schwierig, da man sich auf unterschiedliche Themen konzentrieren muss. Der Grund warum wir das hier schreiben ist, dass der Leser sich dieser Sachlage bewusst ist und weiß, wie in diesem Handbuch versucht wird, beide Themen in einem Wiki zu vereinen. Das Grundlagenwissen wird für gewöhnlich in einem oder zwei kurzen Absätzen am Anfang eines Kapitels besprochen. Der Rest des Handbuchs bezieht sich auf die spezifischen Eigenschaften und Funktionen von Blender. Das Benutzerhandbuch kann keine umfassende Anleitung zu CG-Begrifflichkeiten sein - dieses Wissen sollte durch Bücher, Zeitschriften, Tutorials und manchmal lebenslange Erfahrung erworben werden. Man kann Blender benutzen, um einen kompletten Spielfilm zu erstellen, aber das Lesen dieses Handbuchs wird aus niemandem einen zweiten Steven Spielberg machen.

Ganz abstrakt kann man sich die Benutzung von Blender als Wissen über Fertigkeiten in drei Dimensionen vorstellen:

1. Integration - Rendern von CG-Bildern, Arbeiten mit Realfilmsequenzen oder das Kombinieren dieser beiden (CGI und VFX).
2. Animation - Posing (wichtiger Teil der Charakteranimation) oder allgemein Verformung von Objekten, ob manuell oder durch Simulation.
3. Zeitdauer - Produziert man ein Einzelbild, einen Videoclip, einen Werbespot, einen 10-minütigen Kurzfilm oder einen kompletten Spielfilm?

Fähigkeiten wie die Navigation im 3-dimensionalen Raum, Modelling, Ausleuchtung, Shading, Compositing usw. werden in jedem dieser Bereiche gebraucht. Bessere Kenntnisse darin machen die Arbeit produktiver. Die Werkzeuge von Blender können auch so eingeordnet werden. So hat z.B. der Videoeditor (*Video Sequence Editor*, VSE) recht wenig mit der Fertigkeit der Animation zu tun, ist aber tief in den Achsen Integration und Dauer verwurzelt. Es ist allerdings interessant anzumerken, dass Animationskurven (sog. lpo-Kurven) im VSE benutzt werden, um Effekte zu animieren.

Die Ecken dieses Wissensraumes sind es, an deren die Interessen der meisten Leute liegen; ihr Zielpunkt sozusagen. So gibt es viele talentierte Künstler, die statische CG-Bilder erstellen. Tony Mullens Buch *Introducing Character Animation With Blender* geht um die Verformung von Models mittels Armatures, um eine 1-minütige Animation zu erstellen. Wird die Flüssigkeitssimulation in einem Werbespot verwendet, entspricht dies der Kreuzung von Shape/Sim-Integration-Commercial (siehe Abbildung). "Elephants Dream" und "Big Buck Bunny" bilden eine Blase im Bereich Armature-CG-Integration. Somit sind, abhängig davon was das Endprodukt sein soll, verschiedene Themen und Werkzeuge in Blender von größerem oder kleinerem Interesse.

Eine vierte Dimension bildet das Spieledesign, da es all diese Konzepte vereint und noch das Spielen darüberlegt. Ein Spiel hat nicht nur 1-minütige Zwischensequenzen, sondern auch die eigentliche Spielmechanik, Programmierung der Geschichtenerzählung usw. Das erklärt auch warum es so schwer ist, ein gutes Spiel zu entwerfen; man muss sich in allen Bereichen gut auskennen bevor man das eigentliche Spiel erstellt. Daher wird in diesem Handbuch die Game-Engine nicht näher beleuchtet - dafür gibt es ein eigenes Wikibuch.

Einführung

Seit Version 2.5 hat Blender phänomenale Änderungen in nahezu allen Kategorien erfahren. Darunter: Software, Interface, Modellierung, Animationsfluss, Werkzeuge, die Python API usw. Dies ist das Ergebnis eingehender Analysen von Use Cases (Anwendungsfällen), jahrelangem Hinzufügen von Erweiterungen und Gruppenarbeit, und kompletter Reorganisation und neu Schreiben des Quellcodes der Software. Insgesamt ist dies bis heute eines der größten Projekte, die je an der Code-Basis von Blender unternommen wurde.

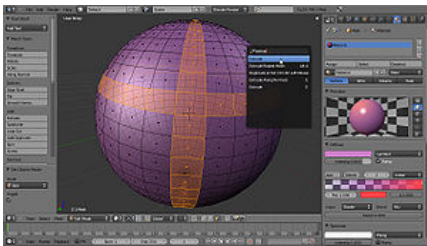
Diese Seite erklärt die größten Unterschieden zwischen Blender 2.4 und Blender 2.5. Dies ist keine ausführliche Liste neuer Funktionalitäten (diese würden Rahmen sprengen!), jedoch ist es eine kurz gefasste Einführung in die Entwicklung von Blender 2.5 und dessen wesentlichen Verbesserungen gegenüber vorherigen Versionen.

Im Folgenden ist ein Video zu sehen, welches in englischer Sprache die wesentlichen Punkte zusammenfasst.

[\[video link\]](#)

Interface (Schnittstelle)

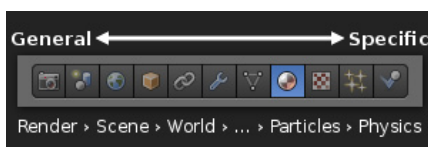
Das neue Benutzer-Interface



Das Blender Benutzer-Interface (UI) basiert auf 3 Prinzipien:

1. **Nicht überlappend** : Das UI erlaubt es alle relevanten Optionen und Werkzeuge auf einem Blick zu sehen, ohne Fenster schieben oder ziehen zu müssen.
2. **Nicht blockierend** : Werkzeuge und Interface-Optionen blockieren den Nutzer bei der Arbeit mit anderen Komponenten von Blender nicht. Blender zeigt keine Pop-Up-Dialoge, die den Nutzer um die Eingabe weiterer Informationen auffordern bevor etwas ausgeführt werden kann.
3. **Nicht Modal** : Benutzereingaben sollten als möglichst konsistent und vorhersehbar verbleiben ohne dabei gewöhnlich verwendete Methoden (wie Maus oder Tastatur) verändern zu müssen.

Das Benutzer-Interface wurde reorganisiert. Alte *Schaltflächen-Fenster* sind jetzt **Eigenschaften**. Eigenschaften zeigen dem Benutzer Daten. Alles, was die Eigenschaften zeigen kann animiert, ausgefahren und vom Anwender willkürlich verändert werden. Das heisst, dass hier keine Werkzeuge sind. Diese sind zur neuen **Werkzeugleiste** der verschiedenen Editoren (wie der 3D Ansicht) gewandert.



Angefangen bei der obersten Ebene enthalten die Eigenschaften eine Liste von Reitern. Die Liste selbst ist analog zur Leserichtung organisiert, sodass die gewöhnlichsten Steuerungen links erscheinen (bzw. Render Eigenschaften), während feinere Steuerungen (bzw. Objekt>Masche>Material>Textur) rechts erscheinen. Weiterin sind wählbare Reiter von der Selektion abhängig (bzw. sind Maschen-Optionen zu Kamera-Optionen unterschiedlich).

[Mehr über neue UI Design Regeln \(englisch\) »](#)

[Mehr über Version 2.5 UI Paradigmen \(englisch\) »](#)

[Mehr über die Eigenschaften Leiste \(englisch\) »](#)

Mehrere Bildschirme

Mit dem neuen Fenstermanager erlaubt Blender die Konfiguration mehrerer Fenster/Bildschirme, was für Aufstellungen mit mehreren Bildschirmen hilfreich ist. Mit dem Hauptfenster kann jedes weitere Fenster in Areale unterteilt werden.

Individualisierbarkeit

```

33
34 class BONE_PT_context_bone(BoneButtonsPanel):
35     bl_label = ""
36     bl_show_header = False
37
38     def draw(self, context):
39         layout = self.layout
40
41         bone = context.bone
42         if not bone:
43             bone = context.edit_bone
44
45         row = layout.row()
46         row.itemL(text="", icon='ICON_BONE_DATA')
47         row.itemR(bone, "name", text="")
48

```

Das UI ist flexibler, als es noch in der Version 2.4x war. Dank der neuen Python-API ist es möglich das Interface zu individualisieren und die Platzierung von Leisten und Schaltflächen zu verändern. Ein großer Teil des Interfaces verwendet Python-Skripten, welche sich im Pfad `./blender/scripts/ui/` befinden. Damit können diese editiert oder neue, eigene Interfaces erstellt werden.

Dank der neuen Python API ist es für Entwickler einfacher Skripten (wie die Render Engine, Werkzeuge, Import/Export Skripten, usw.) in das Blender Interface zu integrieren.

[Read more about new python API »](#)



Weiterhin bietet Blender 2.5 einen neuen **Tastaturkürzel Editor**. Tastaturkürzel oder Maus Definitionen sind zusammen in Tastaturkürzel Konfigurationen gruppiert. Sowie für jeden Editor in Blender, als auch für alle Modi oder Modal Werkzeuge wie Transformation, gibt es mehrere Tastaturkürzel Konfigurationen. Das Individualisieren der Kürzel wird durch ein lokales Kopieren der Standard Konfiguration und daran anschließendem Editieren nach eigenem Gusto realisiert. Die Standard Konfigurationen wird immer unverändert zur Verfügung stehen.

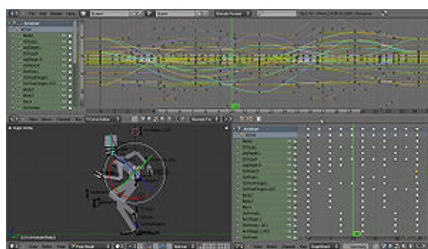
Animations System

Alles ist animierbar!

In Blender 2.5 kann jede Eigenschaft von der Ausgabegröße des Bildes bis hin zu Modifizierungsoptionen animiert werden. Jetzt ist es möglich Markierungen in jedem Editor (wie 3D Ansicht, Videosequenz Editor, Knoten Editor (Material, Textur, Komposite/Zusammensetzung)) zu setzen. Dieses neue System wird *Animato* genannt.

[Mehr über Animato \(englisch\) »](#)

Begleittext- und Grapheneditor



Der IPO Kurveneditor, Aktionseditor und NLA-Editor wurden zum **Begleittext-** und **Grapheneditor** neu zusammengesetzt (generischer Name wird auch in Maya verwendet)

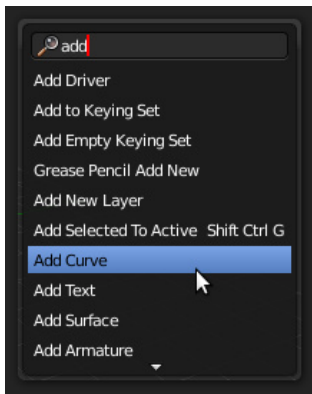
Der "Aktionseditor" wurde zu einem vollständigen Begleittexteditor verändert, welcher die Kontrolle über mehrere Aktionen zugleich, Gruppierung über den Typ und besseren Zugang zu Formen-Kürzeln erlaubt.

Blenders neues Animationssystem erlaubt auch das Hinzufügen von Funktionskurven zu jeder Eigenschaft. Der neue Grapheneditor (förmlich IPO Kurveneditor) erlaubt Ansehen, Durchsehen und Editieren jeder Ansammlung von Funktionskurven inklusive aller Kurven einer kompletten Szene!

[Eine Charakteranimation \(englisch\) »](#)

Neue Funktionen

Suchwerkzeug

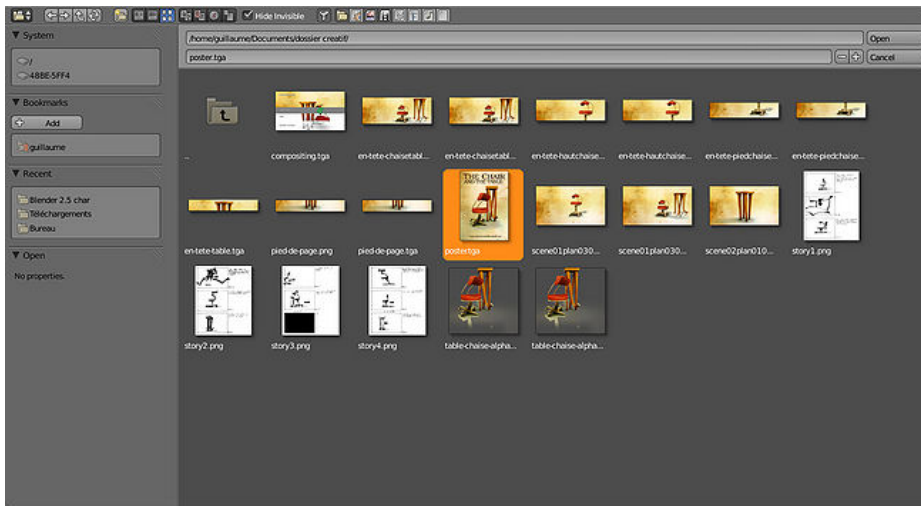


Blender 2.5 integriert ein Suchwerkzeug, welches es erlaubt eine Funktion nach deren Namen (oder einen Teil von dessen) zu finden. Einfach Leertaste egal wo tippen um zu suchen und das Menü wird erscheinen. Das Menü ist auch über dem Blender-Bildschirm verfügbar.

Dateibrowser Verbesserungen

Der alte Dateibrowser und Bildbrowser wurde in einem mächtigen Browser vereint. Dateien können als Liste oder Thumbnails (kleine Vorschaubilder) aufgeführt werden und ein neuer Filter erlaubt die Selektion der gewünschten Dateitypen.

Auch wurde eine Seitenleiste hinzugefügt, in der die Festplatten, die am meisten verwendeten Ordner und Favoriten sichtbar sind.



Python API

Jetzt wird die API basierend auf Python 3.2 verwendet.

Diese Seite als Video!

Diese Seite wurde als ein Video in englischer Sprache auf YouTube umgesetzt.

[\[video link\]](#)

Installieren der Binärdateien

Blender 2.71 steht sowohl als Paket von binären Dateien zur Verfügung wie auch in Form von Quelldateien und ist auf der [Blender Foundation](#) Webseite zu finden. Um Blender 2.71 derzeit herunterzuladen, wählt man in der rechten Navigation dieser Seite den Menüpunkt Blender 2.71 aus.

Beim Wiki Online-Handbuch kann im Allgemeinen das Handbuch der aktuellsten Blender-Version auf der Blender Foundation Website verwendet werden (allerdings kann es sein, dass Features der neuesten Version noch nicht vollständig im Wiki Online-Handbuch überarbeitet sind). Wenn Sie eine veröffentlichte Version dieses Online-Handbuches verwenden, wird empfohlen die Blender-Version auf der veröffentlichten Guide CD-ROM zu benutzen. Wenn im folgenden Text "Download" benutzt wird, sollten Leser des Buches stattdessen die Blender-Version auf der CD-ROM verwenden.

Download und Installation der binären Distribution

Die binäre Distribution ist in den nachfolgenden Varianten erhältlich:

- [Windows](#)
- [Linux](#)
- [MacOSX](#)
- [FreeBSD, Irix, Solaris](#)

Es gibt möglicherweise noch inoffizielle Distributionen für weitere Betriebssysteme, aber nachdem sie nicht von der [Blender Foundation](#) unterstützt werden, sollten Wünsche und Fehler direkt deren Herausgebern gemeldet werden.

Es sind Linux-Binärdateien für die Hardware-Architekturen x86 (Intel und AMD Prozessoren) sowie PowerPC verfügbar. Diese liegen jeweils als statisch verlinkbare sowie als dynamische ladbare Bibliotheken vor.

Der Unterschied zwischen der dynamischen und der statischen Variante ist wichtig. In der statischen Variante sind die OpenGL-Bibliotheken bereits eingebunden. Hierdurch wird Blender auf Ihrem System ohne hardwarebeschleunigte Grafik ausgeführt. Falls die dynamische Version nicht ausführbar ist, versuchen Sie daher bitte die statische Version! OpenGL wird in Blender für alle Zeichenoperationen verwendet, auch für die Darstellung der Menüs und Buttons. Diese Abhängigkeit macht eine korrekte und konforme OpenGL-Installation auf Ihrem System zur Notwendigkeit.

Im Allgemeinen wird Blender auf Systemen mit integrierten oder älteren einfachen Grafikkarten nur schlecht oder gar nicht laufen, da diese OpenGL nur schlecht unterstützen. Oft kann man die schlechte OpenGL-Unterstützung solcher Karten durch den Einsatz von Software basiertem OpenGL umgehen: auf Windows-Systemen durch deaktivieren der Grafik-Hardware-Beschleunigung bzw. durch den Einsatz von von MESA OpenGL auf Linux-Systemen.

Durch die Installation werden Ordner und Dateien an zwei verschiedenen Stellen des Dateisystems angelegt. Eine Gruppe von Verzeichnissen ist für das Programm Blender selbst vorgesehen und eine Gruppe ist für Ihre Benutzerdaten gedacht. Zum Anlegen der letzteren werden Administrationsrechte gebraucht. Hier die Liste dieser Ordner:

- .blender - Konfigurationsinformationen (zum größten Teil in Ihrer Sprache)
- blendcache_.B - temporärer Platz für Informationen rund um die Physics simulation (softbodies, Kleidung, Flüssigkeiten)
- plugins - erweiterte Funktionalitäten für Texturen und Sequencing
- scripts - Für Python Skripte, die die Funktionalität von Blender erweitern
- tmp - Für temporäre Ausgaben und Zwischenstände beim Rendern

Das Rendern wird von der Blender Rendering-Engine im Hauptspeicher und mit dem Hauptprozessor Ihres Systems durchgeführt. Sie können also durchaus eine Maschine mit einer nicht unterstützten Grafikkarte nutzen, wenn Sie diese nur für das Rendering verwenden wollen (wie es z.B. auch auf Rendering-Farmen der Fall ist).

Hardware Unterstützung

Blender unterstützt 64-bit Hardware Plattformen. Auf 64-bit Linux Systemen und Windows 7 (?) wird die Einschränkung auf 2GB adressierbaren Hauptspeicher aufgehoben.

Blender unterstützt auch Multi-CPU/Kern Chips wie Intel Core-Duo and AMD X2 Chips. Es gibt die Einstellung Threads im Performance Abschnitt der Renderoptionen, um die Anzahl der parallel verwendeten Kerne anzuzeigen oder einzustellen. Die Voreinstellung ist der Modus Auto-detect, sodass alle verfügbaren Kerne der CPU für das Rendern verwendet werden, während über den Modus Fixed der Benutzer die Anzahl selbst festlegen kann.

Blender unterstützt eine breite Vielfalt von Grafiktablets auf den gängigen Betriebssystemen, insbesondere auf Linux, OS X und Windows.

Wer die Renderzeiten verkürzen möchte, der findet entsprechende Informationen im Abschnitt [Render](#) dieses Handbuchs.

Entwickler Plattformen

This is the list of systems in use and supported by active Blender developers:

Name	OS	CPU	Graphics card
Andrea Weikert	Windows XP 32	AMD Athlon 64 X2	Nvidia Quadro FX1500
Andrea Weikert	Windows XP 32	Intel P4	ATI Radeon 9000
Andrea Weikert	Linux 32	AMD Athlon 64 X2	Nvidia Quadro FX1500
Antony Riakiotakis	Ubuntu 14.04	Intel Core i5	Nvidia Geforce GT 540M

Antony Riakiotakis	Windows 7 64	Intel Core i5	NVidia Geforce GT 540M
Bastien Montagne	Debian Testing 64	Intel Core i7 Q4700m	NVidia Geforce GTX 850M
Benoit Bolsee	Windows XP 32	AMD Athlon XP	ATI Radeon 9200
Brecht van Lommel	Linux 64	Intel Core 2 Duo	NVidia GeForce 460 GTX
Brecht van Lommel	OS X 10.6	Intel Core 2 Duo	NVidia GeForce 9600M GT
Brecht van Lommel	Windows 7 64	Intel Core 2 Duo	NVidia GeForce 460 GTX
Campbell Barton	Linux 64	AMD-FX 6-Core	Nvidia GeForce GTS 450
Daniel Genrich	Windows Vista 64	Intel Core 2 Duo	NVidia GeForce 8500 GT
Diego Borghetti	Linux 64	Intel Core i5	Nvidia GeForce GTX 480
Diego Borghetti	Linux 64	Intel Core i7	Nvidia GeForce GTX 460M
Dustin Martin	OSX 10.5	Dual Quad Intel	Nvidia Geforce 8800 GT
Howard Trickey	Ubuntu 12.04 64	Intel Xeon E5-1650	NVidia Quadro 600
Howard Trickey	Windows 7 64	Intel Core i7	NVidia GeForce GTX 460
Howard Trickey	OSX 10.8.2	Intel Core Duo	NVidia GeForce 9400M
Jens Verwiebe	OSX 10.6/7/8/9	Intel Xeon 6-core@ 3.33	ATI 5870/7970
Jeroen Bakker	Latest Ubuntu 64bit	Dell m4300 Intel Core 2 Duo 2.0Ghz	Nvidia Quadro FX360M
Joshua Leung	Windows Vista 32	Intel Core2 Duo	Nvidia GeForce Go 7600
Julian Eisel	Linux 64	AMD Phenom II X4	Nvidia GeForce GTX 570
Julian Eisel	Linux 64	Intel Core i7	Nvidia GeForce GT 645M
Ken Hughes	Linux 32	Intel Core Duo	Nvidia GeForce GO 7500
Ken Hughes	Linux 64	AMD Athlon 64 X2	Nvidia GeForce 6600
Kent Mein	Linux 64	Intel Core Duo	Nvidia Quadro FX 1400
Kent Mein	SunOS 5.8	Sun Blade 150	ATI PGX
Matt Ebb	OSX 10.5	Dual Core Intel MBP	nVidia 8600M
Michael Fox	Linux 32	Celeron	Nvidia GeForce 6200
Nathan Letwory	Windows 7 Ultimate 64	AMD Turion X2 Mobile RM-74	ATI HD 4650
Nathan Letwory	Windows 7 Ultimate 64	AMD Athlon II X4 620	2x HIS ATI HD 5550 /w four monitors
Nicholas Bishop	Fedora 18 64bit	Intel Core i7 @ 2.93GHz	AMD Radeon HD 6950 (Gallium drivers, currently at OpenGL 2.1)
Nicholas Bishop	Ubuntu 12.10 64bit	Intel Core i5	ATI Mobility Radeon 5650 (Gallium drivers)
Raul Fernandez Hernandez	Linux 32	Pentium D 945	ATI X1550
Robin Allen	Linux 32	Intel Centrino duo	NVidia GeForce go 7600
Robin Allen	Windows XP 32	Intel Centrino duo	NVidia GeForce go 7600
Sergej Reich	Arch Linux 64bit	Intel Core2 Quad @ 2.83GHz	Nvidia GeForce GTX 285
Sergej Reich	Arch Linux 64bit	Intel Core i3 @ 2.10GHz	Intel Sandybridge Mobile
Sergey Sharybin	Debian Wheezy 64bit	Intel Core i7 920 2.6Ghz	Nvidia GeForce GTX 560Ti + GeForce GT 620
Sergey Sharybin	Debian Wheezy 64bit	Intel Core i5 2.4GHz	Intel Sandy Bridge + Nvidia GT 520M
Thomas Dinges	Windows 7 x64	Intel Core i5	Intel HD 2500
Thomas Dinges	Windows 7 x64	Intel Core i7	NVidia GeForce 540M + Intel HD 3000
Timothy Baldridge	SGL Irix 6.5 (mipspro)	8 x R16000	(headless)
Timothy Baldridge	SGL Irix 6.5 (mipspro)	2 x R10000	
Tamito Kajiyama	Windows Vista 64bit	Intel Core2 Duo	Nvidia Quadro FX 770M
Ton Roosendaal	OSX 10.7	iMac Intel Core i7	AMD Radeon HD 6970M

Ton Roosendaal	OSX 10.8	MacBook Pro i7 "Retina"	NVidia GT 650M + Intel HD 4000
----------------	----------	-------------------------	--------------------------------

Übersetzen der Quellen

Gegenwärtig gibt es vier verschiedene Build-Systeme, um die ausführbaren Dateien für die verschiedenen, unterstützten Betriebssysteme zu bauen. Schlagen Sie unter [Building Blender](#) nach, um weitere Informationen zum Kompilieren eines anwenderspezifischen Setups für Ihre Hardware zu erhalten.

Übersetzen der Plugins

Plugins sind dynamisch geladene Routinen, die die Funktionalität von Blender in verschiedenen Bereichen wie der Generierung von Texturen oder dem Sequencing (Bild/Video Gestaltung) erweitern. Nähere Informationen dazu findet ihr in den [Forenbeiträgen](#) von <http://blenderartists.org/>.

Die Skriptsprache Python

[Python](#) ist eine universelle Skriptsprache, für welche in Blender eine spezielle Schnittstelle vorgesehen wurde, damit alle internen Funktionen von Blender von dieser Sprache angesprochen werden können. Skripten, die in dieser Sprache entwickelt werden, können die Funktionalität von Blender erweitern, ohne dass Blender deshalb neu übersetzt werden muss. Diese Programme werden von Anwendungsprogrammierern erstellt.

Die erwartete Version von [Python](#) ist normalerweise Bestandteil des jeweiligen Blender-Setups und kommt durch das Installieren auf den Rechner. Allerdings kann die gewünschte Python Version auch direkt von der offiziellen [Python Webseite](#) geholt und installiert werden. Dabei sollte jedoch auf die Version von Python geachtet werden, da die Python-Version, mit der Blender erstellt wurde, mit dieser in den ersten beiden Stellen übereinstimmen muss (Bei Blender 2.67 ist es "3.3").

Wie könnt ihr feststellen, welche Python Version von Blender erwartet wird?

Startet dazu Blender und öffnet die Python-Console. Gebt dort `help` oder `help()` ein und bei einer dieser Varianten erhaltet ihr die Version.

Unter Linux wird Euch viel Arbeit abgenommen, sobald Python installiert ist. könnt ihr sofort loslegen.

Unter Windows kann es sein, dass ihr erst Eure Umgebung festlegen müsst.

<Abschnitt für Windows-Benutzer (kann nichts dazu beitragen, ob noch aktuell ...)>

Most functions do not rely on [Python](#) – a notable exception is the Help menu, which opens a web browser pointed to a specific location. Help text is not bundled into Blender; you must download the latest wiki or PDF user manuals, found [here](#) or at www.blender.org.

In general, wherever you install Python, you need to establish an operating system environment variable `PYTHONPATH` and point it to the Blender Scripts directory where python modules are installed, e.g. `C:\Program Files\Blender Foundation\Blender\scripts\bpymodules` for Windows machines. Environment variables on Windows machines are set in the advanced Systems settings, in the Control Panel.

</Abschnitt für Windows-Benutzer>

Wird Blender auf einer Maschine mit einem richtig installierten Python gestartet, dann erhaltet ihr im Python-Console Fenster eine Meldung. Unter Linux sieht sie etwa so aus:

```
PYTHON INTERACTIVE CONSOLE 3.3.0 (default, Nov 6 2012, 11:42:41) [GCC 4.7.1]Command History:      Up/Down Arrow
Cursor:      Left/Right Home/End
Remove:      Backspace/Delete
Execute:      Enter
Autocomplete: Ctrl-Space
Zoom:      Ctrl +/-, Ctrl-Wheel
Builtin Modules:  bpy, bpy.data, bpy.ops, bpy.props, bpy.types, bpy.context, bpy.utils, bgl, blf, mathutils
Convenience Imports: from mathutils import *; from math import *
Convenience Variables: C = bpy.context, D = bpy.data
```

Unter Windows/MAC anders? Dann ergänzen ... ;-)

Diese Meldung bedeutet, dass Python installiert ist, die Umgebung für Entwicklung und Ausführung gesetzt ist und der Zugriff sowie die Ausführung aller Python-Skripte, die bei Blender mitgeliefert wurden oder für Blender zur Verfügung stehen, gewährleistet ist.

Sollten stattdessen andere Meldungen kommen, wie etwa

```
Could not find platform independent libraries <prefix>
Could not find platform dependent libraries <exec_prefix>
Consider setting $PYTHONHOME to <prefix>[:<exec_prefix>]
'import site' failed; use -v for traceback
Checking for installed Python... No installed Python found.
Only built-in modules are available. Some scripts may not run.
Continuing happily.
```

dann ist Python nicht vollständig. Falls die vollständige Funktionalität von Python gewünscht ist, dann auf der [Python Webseite](#) den Installationsanweisungen folgen.

When you install Blender, you must tell the Python module where you put the scripts. If you choose to put user data in a different location for each user, then the install will put your scripts in the `C:\Users\<Current User>\AppData\Roaming\Blender Foundation\Blender\blender\scripts` folder. If you are upgrading, you probably want to overwrite all your old scripts with the new versions, and not have several versions of the same script hanging around on your PC. The best place, if you will not be editing them, is to put them in your Program Files folder with Blender:

1. Do a search on your machine for a file name with the word "scripts".
2. You will see the scripts folder appear after the initial search: `C:\Program Files\Blender-2.49\blender\scripts`, or something similar...
3. Open the script folder from the search window. You will see all the scripts. You can leave them there, or put them on your desktop temporarily...
4. Then go to `Program Files`, then to `Blender Foundation`, then `blender` folder, then make inside it a new folder called `scripts`...
5. Drag and drop or copy all the scripts from where ever you put them into this folder.
6. Make sure to include the two module folders in the script one.
7. Then, if you don't know this already, open Blender.
8. In Blender, the top menu bar hides all the preferences. Drag it down and then you will see a button marked `File Paths`.
9. Once you click that `File Paths` button, a set of path fields will be revealed.

10. Go to the Python Scripts one and type in the script folder path you just created (or use the “folder” button to open a file browser – and hit the SELECT SCRIPT PATH button to validate).
11. Then go to the File menu and Save Default Settings (CtrlU), so that Blender will remember that the script folder is where you told it to look!
12. Be careful though if you have already done stuff in Blender at this point, every time you start it it will be the default start up...

Installation auf Windows

Download

Sie erhalten die letzte stabile Version von Blender für Windows auf der [Blender Downloadseite](#).

Version

Blender für Windows ist zurzeit in der 32-bit- und 64-Bit-Versionen verfügbar. Benutzer mit einer 32-Bit-Version von Windows müssen die 32-Bit-Version von Blender herunterladen. Benutzer mit einer 64-Bit-Version von Windows haben die Wahl, entweder die 32-bit- oder 64-Bit-Version von Blender zu verwenden. Jedoch ist die 64-bit-Version von Blender auf Systemen mit viel RAM leistungsfähiger.

Um festzustellen ob Sie eine 32-bit- oder 64-Bit-Version von Windows haben, können Sie auf der C:\ Partition nachsehen. Wenn es einen *Program Files (x86)* Ordner gibt, dann haben Sie eine 64-Bit-Version von Windows. In dem *Program Files (x86)* Ordner befinden sich die 32-Bit-Programme.

Installation

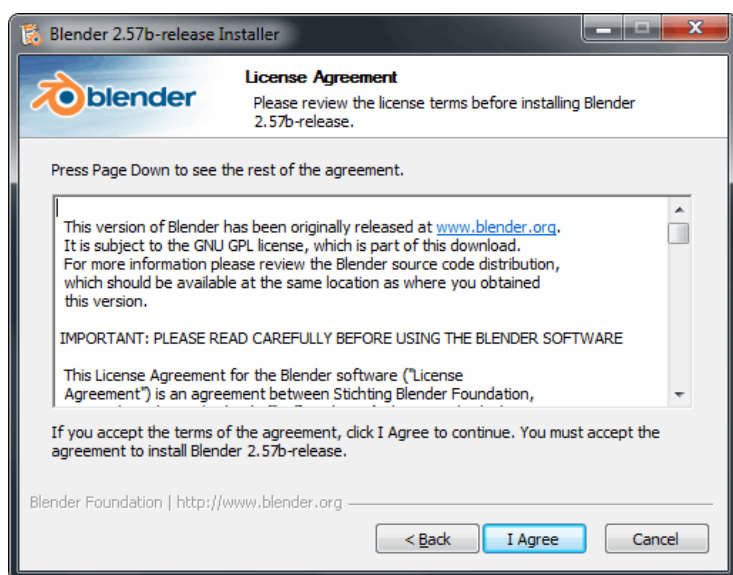
Sobald der Download fertig ist, wechseln Sie in Ihren 'Download'-Ordner und klicken doppelt auf die ausführbare Blender Datei, um den Installationsprozess zu beginnen. Beachten Sie, dass die Installation von Blender Administratorrechte benötigt.

Willkommenbildschirm



Der Willkommensbildschirm ist der Erste des Installationsprozesses. Klicken Sie auf 'Next' um fortzufahren.

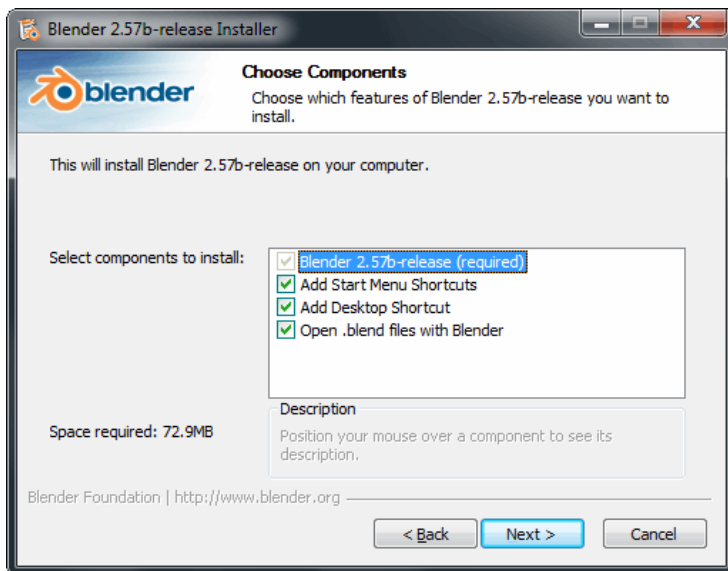
Lizenzvertrag



Der Lizenzvertrag muss akzeptiert werden, bevor die Installation fortgesetzt werden kann.

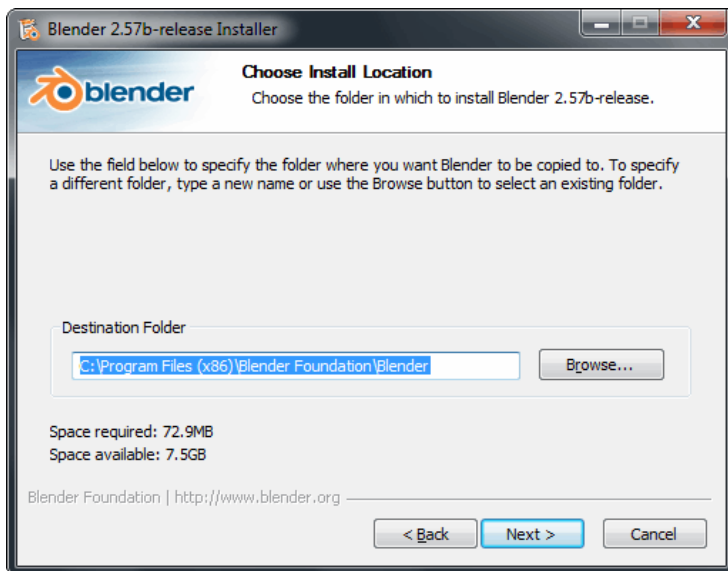
Installationsoptionen

Programmoptionen



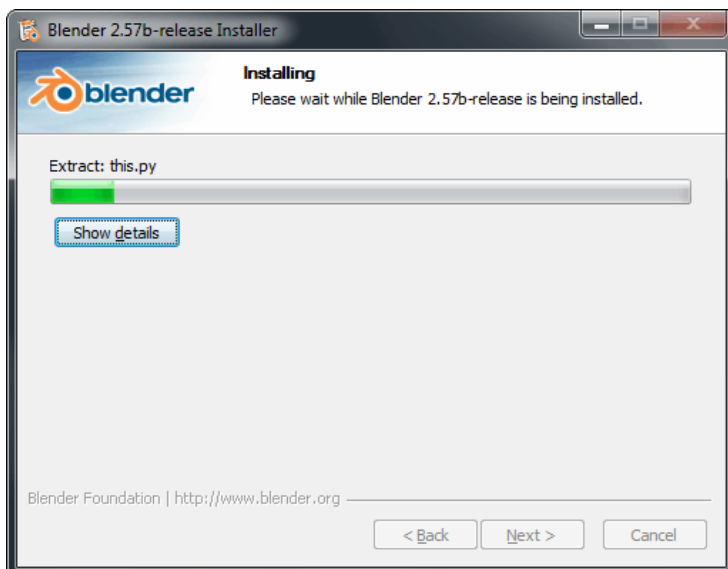
Wählen Sie die gewünschten Optionen aus.

Verzeichnis

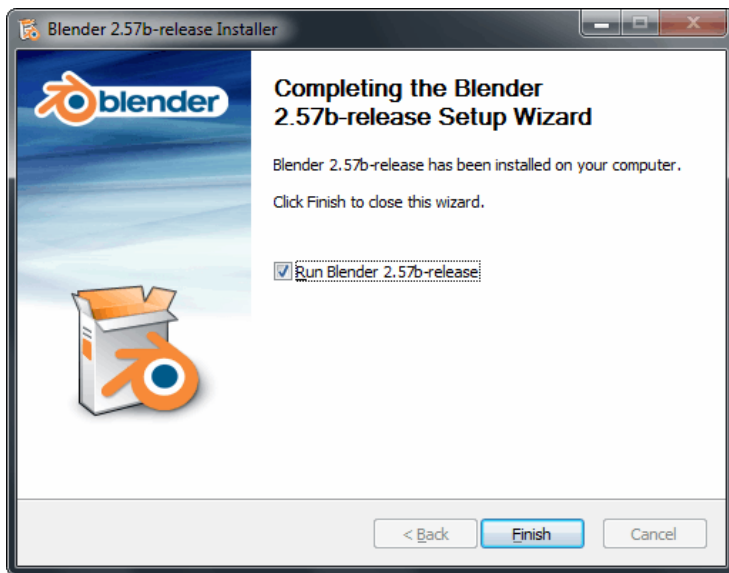


Hier können Sie das Verzeichnis wählen, in dem Blender installiert werden soll.

Installieren



Installiert



Und Fertig! Eine frische Installation von Blender 2.6. Nach dem Sie eingestellt haben ob Blender nach der Installation gestartet werden soll, klicken Sie auf 'Finish'.

Tragebare Installation

Wenn Sie Blender von einem USB-Stick ohne Installation ausführen wollen, müssen Sie die .zip Version auf einen USB-Stick extrahieren. You may want to avoid storing animation output or other temporary files on the USB key as frequent drive writes may shorten its life.

Um alle Konfigurationsdateien und installierte addons auf dem USB Stick zu behalten, erstellen Sie einen *config* Ordner in dem entpackten Blender-Ordner. Jetzt werden alle Konfigurationsdateien von dort gelesen und geschrieben, jedoch nicht mehr auf dem Computer, auf dem Sie Blender ausführen.

Installieren auf Linux

Download

Die neueste stabile Version von Blender für Linux bekommen Sie bei der [Blender Downloadseite](#) oder aus dem Software-Repository Ihrer Distribution, wenn diese ein Paket für Blender bereitstellt (z.B. Ubuntu-Software-Center).

Version

Blender für Linux ist derzeit in 32-bit und 64-bit-Versionen verfügbar. Benutzer mit einer 32bit-Version von Linux müssen die 32-bit-Version von Blender nutzen. Benutzer mit einer 64-bit-Version von Linux können entweder die 32-bit oder die 64-bit-Version wählen, jedoch werden Sie wahrscheinlich eine Geschwindigkeitsverbesserung bemerken, wenn Sie die 64-bit-Version von Blender verwenden, besonders bei Systemen mit einer großen Menge RAM.

Um zu bestimmen, ob Sie eine 32-bit oder eine 64-bit-Version von Linux haben, können Sie entweder in Ihrer Distributionsdokumentation nachschauen oder den `uname` Befehl mit der `-m` Option nutzen. `uname` gibt Systeminformationen aus und die `-m` Option zeigt die Hardwarearchitektur an.

- Öffnen des Terminal
- Eingeben des Befehls `uname -m`

Wenn Sie ein 32-bit-System haben, wird `uname -m` den Wert `i686` zurückgeben. Bei einem 64-bit-System wird dieser Wert `x86_64` sein.

Distributionsversionen

Die meisten Hauptdistributionen wie Ubuntu, Debian, OpenSUSE, Fedora und viele andere werden einen Build von Blender in ihrem Software Repository anbieten, welcher über den Paketmanager abgerufen werden kann. Wenn Ihre Distribution dass nicht tut, oder ihr Repository nicht auf die neueste Blenderversion aktualisiert hat, können Sie Blender mit den unten angeführten Anweisungen selbst installieren.

Denken Sie daran, dass Distributionsversionen meistens veraltet sind.

Installation

Kontrollieren ob Ihre Distribution die neueste Blenderversion in ihrem Paketmanager anbietet. Wenn nicht, herunterladen der passenden Version von Blender für Linux von der [Blender-Downloadseite](#) und entpacken des Archives zu einem Ort Ihrer Wahl.

Dies wird einen Ordner mit dem Namen `blender-VERSION-linux-glibcVERSION-ARCH` erstellen, wobei `VERSION` die Blenderversion, `glibcVERSION` die Version des benötigten glibc ist und `ARCH` ist deine Computerarchitektur (`i686` or `x86_64`). In diesem Ordner finden Sie die `blender` Binärdatei.

Um Blender zu starten,


- Starten Ihres [X.Org Server](#) (wenn er nicht bereits wie bei den meisten Distributionen standardmäßig läuft)
- Navigieren zum Blenderordner mit einem Dateimanager und Doppelklick auf die ausführbare Blenderdatei oder,
- Öffnen der Konsole, navigieren zum Blender Ordner und ausführen des Befehls `./blender`

In `/opt` installieren oder unter `/usr/local`

Sie können Blender auch in `/opt` oder in `/usr/local` installieren, indem Sie den Blender Ordner in einen dieser Orte verschieben. Wenn Sie Blender von jedem Ordner aus ausführen möchten, müssen Sie auch die Pfadvariable aktualisieren. Schauen Sie dafür in der Dokumentation für Ihr Betriebssystem für die empfohlene Methode nach.

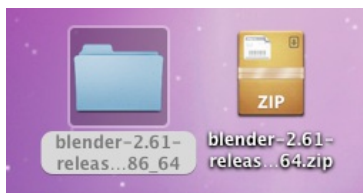
Konfiguration

Alt+Mouse Konflikt

Einige Linuxdistributionen verwenden standardmäßig `Alt LMB`  um Fenster zu verschieben, da aber Blender `Alt+Klick` verwendet ist es normalerweise am Einfachsten diese Funktion zu deaktivieren oder die Taste auf die Super-Taste zu legen (Windows-taste).

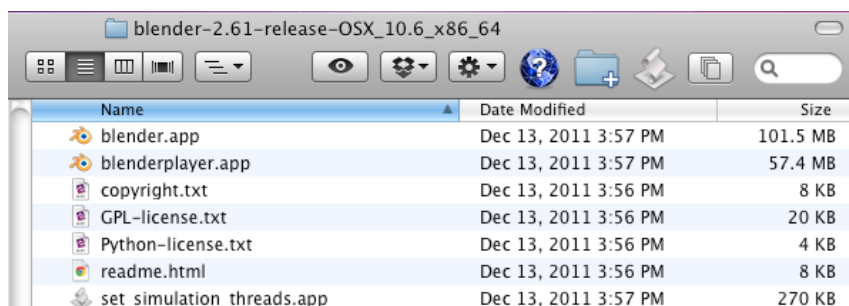
Installation auf Mac

Blender auf einem Mac zu installieren ist sehr einfach. Es sind vorkompilierte Version für PowerPC-Prozessoren und 32 und 64-bit-Macs mit Intel-Prozessoren vorhanden. Wenn Sie die Plattform und den Prozessor Ihres Mac kennen:



Blender verpackt (rechts) und entpackt (links)

- Laden Sie die für Ihr System entsprechende Blender Version auf der [Blender-Downloadseite](#) herunter.
- Rechtsklick RMB auf die heruntergeladene Datei, und *Zu einem Ordner entpacken* wählen.
- Es entsteht ein Ordner, wo Sie Blender heruntergeladen haben, mit demselben Namen der heruntergeladenen Datei, aber ohne Dateinamen-Erweiterung.
- Klicken Sie auf den Ordner und er wird im Finder geöffnet (Bild: Blender vom Finder öffnen - siehe unten)
- Führen Sie einen Doppelklick LMB auf die *blender.app* Datei aus. Es kann losgehen!



Blender vom Finder öffnen - Beispiel mit Blender 2.61 für einen 64bit-Intel Mac, Mac OS X 10.6

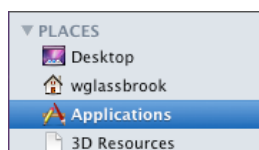


Unterstützte Systeme für Blender 2.6X und darüber

Blender 2.6X-Versionen und darüber werden nur auf Systemen mit einem PowerPC G5 Prozessor oder Macs mit Intel-Prozessoren laufen, Mac OS X 10.4 oder höher wird benötigt.

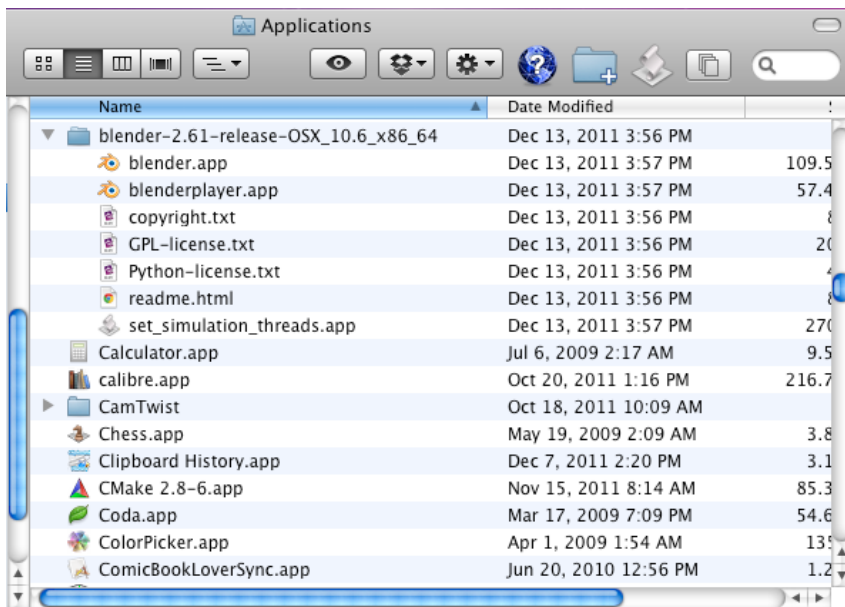
Wichtig: Wenn Sie Blender von der Kommandozeile ausführen müssen um das Konsolenfenster zu sehen, siehe Seite [Blender-Konsole](#)

Blender zu den Anwendungen hinzufügen



Orte und Anwendungen auf Mac


Sie können Blender auch zum Anwendungsordner von Mac OS X verschieben. Es funktioniert wie bei anderen Anwendungen auf Ihrem Mac. Dies erreichen Sie, in dem Sie den entpackten Ordner zum *Anwendungen*-Ordner kopieren oder verschieben. Wählen Sie den Ordner *Anwendungen* mit dem Finder-Dateimanager und suchen Sie nach *Orte*. Auf der linken Seite öffnen Sie den Tab *Orte* und werden den Ordner *Anwendungen* finden. Verschieben Sie den entpackten Ordner von Blender dann zu den Anwendungen.



Blender nach dem Verschieben zu den Anwendungen

Blender zum Dock hinzufügen

Um Blender zum Dock hinzuzufügen, führen Sie folgendes aus:

- Wenn Blender im *Anwendungen*-Ordner liegt, klicken Sie auf *blender.app* und verschieben es zum Dock.
- Wenn Blender läuft, Rechtsklick RMB  auf das vorhandene Symbol im Mac Dock und Auswählen von *Im Dock halten*'.

Andere unterstützte Betriebssysteme

FreeBSD

Lade die Datei `blender-2.##-FreeBSD-####.tbz` von der [Blender Downloadseite](#) herunter, wobei 2.## die Blenderversion und #### die Hardwarearchitektur ist (i386 oder amd64).

Blender starten

- Archiv entpacken
- Konsole öffnen und zum entpackten Ordner navigieren
- Den Befehl `./blender` ausführen während der [X.Org Server](#) läuft.

Andere nicht unterstützte Betriebssysteme

MorphOS

Portierungen für diese Plattform werden durch Guillaume Roguez unterstützt. Mehr Informationen zu dieser Portierung gibt es bei [seiner Wiki \(französische Version\)](#).

Die Installation dieser Portierung fordert das Entpacken zu einem Ort auf Deiner Festplatte anstatt in den [RAM](#), da Python dann nicht in der Lage ist seine dynamischen .pym-Module im RAM zu verwenden.

Konfiguration & Dateipfade

Blender kann systemweit installiert oder aus einem extrahierten Archiv, welches alle benötigten Dateien enthält, gestartet werden.

Es gibt 3 verschiedene Ordner, welche Blender verwenden wird. Deren exakte Lokation ist betriebssystemabhängig.

- **LOCAL:** Lokation der Konfiguration und Laufzeitdateien (für enthaltene archive)
- **USER:** Lokation der Konfigurationsdateien (normalerweise das Home-Verzeichnis des Benutzers)
- **SYSTEM:** Lokation der Laufzeitdateien für die systemweite Installation (evtl. nur-lesend).

Für Systeminstallationen die Verzeichnisse **SYSTEM & USER** benötigt.

Für ein lokal extrahiertes Blender werden die Benutzerkonfiguration und Laufzeitdaten im selben Unterverzeichnis gehalten, was es erlaubt mehrere Instanzen von Blender ohne Konflikte zugleich zu starten, wobei die **USER** und **SYSTEM** Dateien ignoriert werden.

Im Folgenden die Standardlokationen für jedes System:

OSX

LOCAL: ./2.63/

USER: /Users/{user}/Library/Application Support/Blender/2.63/

SYSTEM: /Library/Application Support/Blender/2.63/

zu beachten ist, dass OSX die Blender Binärdaten im Ordner .blender.app/Contents/MacOS/blender abspeichert, somit ist der lokale Pfad zu Dateien und Konfigurationen .blender.app/Contents/MacOS/2.63

Windows

LOCAL: .\2.63\

USER: C:\Documents and Settings\{username}\AppData\Roaming\Blender Foundation\Blender\2.63\

SYSTEM: C:\Documents and Settings\All Users\AppData\Roaming\Blender Foundation\Blender\2.63\

Unix (Linux/BSD/Solaris)

LOCAL: ./2.63/

USER: \$HOME/.blender/2.63/

SYSTEM: /usr/share/blender/2.63/

zu beachten ist, dass ./2.63/ relativ zu Blenders ausführbaren Dateien ist und für enthaltene Archive, welche über die offizielle Seite blender.org verteilt werden, verwendet wird.

Pfad Layout

Im Folgenden wird das Pfadlayout beschrieben, welches für die zuvor aufgeführten Verzeichnisse verwendet wird.

Wobei beispielsweise ./config/**startup.blend** ~/.blender/2.63/config/startup.blend sein kann.

- ./autosave/ ...
Lokation der automatisch gespeicherten .blend-Dateien. *gilt nur für Windows, andere Systeme verwenden das temp-Verzeichnis..*
Suchreihenfolge: **LOCAL, USER**
- ./config/ ...
Standards und Sitzungsinformationen
Suchreihenfolge: **LOCAL, USER**
- ./config/**startup.blend**
standardmäßige Datei, welches beim Start geladen wird.
- ./config/**bookmarks.txt**
Lesezeichen im Dateibrowser.
- ./config/**recent-files.txt**
Liste der zuletzt verwendeten Dateien.
- ./datafiles/ ...
Laufzeitdateien
Suchreihenfolge: **LOCAL, USER, SYSTEM**
- ./datafiles/locale/{language}/
Übersetzungen. *wird derzeit nicht verwendet!*
- ./datafiles/icons/*.png
Icon Themen für Blenders Benutzerschnittstelle (UI). *derzeit nicht in den Themeneinstellungen wählbar.*
- ./datafiles/brushicons/*.png
Bilder für jeden Pinsel.
- ./scripts/ ...
Python Skripten für das UI und Werkzeuge
Suchreihenfolge: **LOCAL, USER, SYSTEM**
- ./scripts/addons/*.py

Python Erweiterungen, welche in den Benutzereinstellungen aktiviert werden können. Enthält die Unterstützung für Dateiimport/-export, Render Engine Integration und viele handliche Hilfsmittel.

- `./scripts/addons/modules/*.py`
Module, die für Erweiterungen verwendet werden können (wurde zu Pythons `sys.path` hinzugefügt)
- `./scripts/addons_contrib/*.py`
ein weiteres Verzeichnis für Erweiterungen, welches für gemeinschaftlich gewartete Erweiterungen bestimmt ist (muss manuell angelegt werden)
- `./scripts/addons_contrib/modules/*.py`
Module die für verteilte Erweiterungen verwendet werden können (wurde zu Pythons `sys.path` hinzugefügt)
- `./scripts/modules/*.py`
Python Module, die unsere Kern-API und Hilfsfunktionen für das Importieren anderer Skripten enthält (wurde zu Pythons `sys.path` hinzugefügt)
- `./scripts/startup/*.py`
Skripten, die beim Programmstart automatisch importiert werden.
- `./scripts/presets/{preset}/*.py`
Voreinstellungen, die für das Speichern von benutzerdefinierten Einstellungen für den Kleidung-Effekt, Render Formate usw. verwendet wird
- `./scripts/templates/*.py`
Beispielskripten, die über den Kopf des Textrahmens -> Text -> Skript Textschablonen erreichbar sind.
- `./python/ ...`
gebündelte Python Distribution, welche nur benötigt wird, wenn das Python des Systems fehlt oder inkompatibel ist.
Suchreihenfolge: **LOCAL**, **SYSTEM**

Anmerkungen

Pfad für Benutzerskripten

Der Pfad für benutzerdefinierte Skripten bietet eine Möglichkeit ein eigenes Verzeichnis zu definieren, welches sowohl für Skripten, als auch als Benutzerskripten Pfad verwendet wird. Dazu müssen Unterverzeichnisse, wie `startup/`, `addons/`, `modules/`, usw., analog zur Struktur Blenders Skriptenverzeichnis angelegt werden, da direkt kopierte Skripten in dieses Verzeichnis beim Start nicht geladen werden.

Umgebungsvariablen

Umgebungsvariablen können für das Überschreiben der Standardpfade verwendet werden, bzw: `$BLENDER_USER_CONFIG`, `$BLENDER_SYSTEM_PYTHON`.

Dieses Vorhaben benötigt normalerweise keine Einstellungen, kann jedoch nützlich für eigene Konfigurationen sein.

Weitere Details können im Abschnitt 'Environment Variables' in 'blender --help' eingesehen werden (englisch).

Skriptenpfade und fehlende Schaltflächen

Wenn Blender ohne UI startet, liegt es eventuell daran, dass die Skripten nicht korrekt geladen wurden. Die Ursachen können sein...

- der Skriptenpfad wurde nicht gefunden.
- ein Fehler in einem Skript.
- ein Versionskonflikt zwischen Blender und den Skripten.

Am Besten startet man Blender in diesem Fall aus einem Terminal/Befehlsfenster, um alle Fehlermeldungen sehen zu können und um festzustellen, wo die Fehlerursache ist.

Konfiguration & Dateipfade

Blender kann systemweit installiert oder aus einem extrahierten Archiv, welches alle benötigten Dateien enthält, gestartet werden.

Es gibt 3 verschiedene Ordner, welche Blender verwenden wird. Deren exakte Lokation ist betriebssystemabhängig.

- **LOCAL:** Lokation der Konfiguration und Laufzeitdateien (für enthaltene archive)
- **USER:** Lokation der Konfigurationsdateien (normalerweise das Home-Verzeichnis des Benutzers)
- **SYSTEM:** Lokation der Laufzeitdateien für die systemweite Installation (evtl. nur-lesend).

Für Systeminstallationen die Verzeichnisse **SYSTEM** & **USER** benötigt.

Für ein lokal extrahiertes Blender werden die Benutzerkonfiguration und Laufzeitdaten im selben Unterverzeichnis gehalten, was es erlaubt mehrere Instanzen von Blender ohne Konflikte zugleich zu starten, wobei die **USER** und **SYSTEM** Dateien ignoriert werden.

Im Folgenden die Standardlokationen für jedes System:

OSX

LOCAL: `./2.63/`

USER: `/Users/{user}/Library/Application Support/Blender/2.63/`

SYSTEM: `/Library/Application Support/Blender/2.63/`

zu beachten ist, dass OSX die Blender Binärdaten im Ordner `./blender.app/Contents/MacOS/blender` abspeichert, somit ist der lokale Pfad zu Dateien und Konfigurationen `./blender.app/Contents/MacOS/2.63`

Windows

LOCAL: `.\2.63\`

USER: `C:\Documents and Settings\{username}\AppData\Roaming\Blender Foundation\Blender\2.63\`

SYSTEM: `C:\Documents and Settings\All Users\AppData\Roaming\Blender Foundation\Blender\2.63\`

Unix (Linux/BSD/Solaris)

LOCAL: `./2.63/`

USER: `$HOME/.blender/2.63/`

SYSTEM: `/usr/share/blender/2.63/`

zu beachten ist, dass `./2.63/` relativ zu Blenders ausführbaren Dateien ist und für enthaltene Archive, welche über die offizielle Seite blender.org verteilt werden, verwendet wird.

Pfad Layout

Im Folgenden wird das Pfadlayout beschrieben, welches für die zuvor aufgeführten Verzeichnisse verwendet wird.

Wobei beispielsweise `./config/startup.blend` `~/.blender/2.63/config/startup.blend` sein kann.

- `./autosave/ ...`
Lokation der automatisch gespeicherten .blend-Dateien. *gilt nur für Windows, andere Systeme verwenden das temp-Verzeichnis..*
Suchreihenfolge: **LOCAL, USER**
- `./config/ ...`
Standards und Sitzungsinformationen
Suchreihenfolge: **LOCAL, USER**
- `./config/startup.blend`
standardmäßige Datei, welches beim Start geladen wird.
- `./config/bookmarks.txt`
Lesezeichen im Dateibrowser.
- `./config/recent-files.txt`
Liste der zuletzt verwendeten Dateien.
- `./datafiles/ ...`
Laufzeitdateien
Suchreihenfolge: **LOCAL, USER, SYSTEM**
- `./datafiles/locale/{language}/`
Übersetzungen. *wird derzeit nicht verwendet!*
- `./datafiles/icons/*.png`
Icon Themen für Blenders Benutzerschnittstelle (UI). *derzeit nicht in den Themeneinstellungen wählbar.*
- `./datafiles/brushicons/*.png`
Bilder für jeden Pinsel.
- `./scripts/ ...`
Python Skripten für das UI und Werkzeuge
Suchreihenfolge: **LOCAL, USER, SYSTEM**
- `./scripts/addons/*.py`
Python Erweiterungen, welche in den Benutzereinstellungen aktiviert werden können. Enthält die Unterstützung für Dateiimport/-export, Render Engine Integration und viele handliche Hilfsmittel.
- `./scripts/addons/modules/*.py`
Module, die für Erweiterungen verwendet werden können (wurde zu Pythons sys.path hinzugefügt)
- `./scripts/addons_contrib/*.py`
ein weiteres Verzeichnis für Erweiterungen, welches für gemeinschaftlich gewartete Erweiterungen bestimmt ist (muss manuell angelegt werden)
- `./scripts/addons_contrib/modules/*.py`
Module die für verteilte Erweiterungen verwendet werden können (wurde zu Pythons sys.path hinzugefügt)
- `./scripts/modules/*.py`
Python Module, die unsere Kern-API und Hilfsfunktionen für das Importieren anderer Skripten enthält (wurde zu Pythons sys.path hinzugefügt)
- `./scripts/startup/*.py`
Skripten, die beim Programmstart automatisch importiert werden.
- `./scripts/presets/{preset}/*.py`
Voreinstellungen, die für das Speichern von benutzerdefinierten Einstellungen für den Kleidung-Effekt, Render Formate usw. verwendet wird
- `./scripts/templates/*.py`
Beispielskripten, die über den Kopf des Textrahmens -> Text -> Skript Textschablonen erreichbar sind.
- `./python/ ...`
gebündelte Python Distribution, welche nur benötigt wird, wenn das Python des Systems fehlt oder inkompatibel ist.

Suchreihenfolge: **LOCAL, SYSTEM**

Anmerkungen

Pfad für Benutzerskripten

Der Pfad für benutzerdefinierte Skripten bietet eine Möglichkeit ein eigenes Verzeichnis zu definieren, welches sowohl für Skripten, als auch als Benutzerskripten Pfad verwendet wird. Dazu müssen Unterverzeichnisse, wie `startup/`, `addons/`, `modules/`, usw., analog zur Struktur Blenders Skriptenverzeichnis angelegt werden, da direkt kopierte Skripten in dieses Verzeichnis beim Start nicht geladen werden.

Umgebungsvariablen

Umgebungsvariablen können für das Überschreiben der Standardpfade verwendet werden, bzw: `$BLENDER_USER_CONFIG`, `$BLENDER_SYSTEM_PYTHON`.

Dieses Vorhaben benötigt normalerweise keine Einstellungen, kann jedoch nützlich für eigene Konfigurationen sein.

Weitere Details können im Abschnitt 'Environment Variables' in 'blender --help' eingesehen werden (englisch).

Skriptenpfade und fehlende Schaltflächen

Wenn Blender ohne UI startet, liegt es eventuell daran, dass die Skripten nicht korrekt geladen wurden. Die Ursachen können sein...

- der Skriptenpfad wurde nicht gefunden.
- ein Fehler in einem Skript.
- ein Versionskonflikt zwischen Blender und den Skripten.

Am Besten startet man Blender in diesem Fall aus einem Terminal/Befehlsfenster, um alle Fehlermeldungen sehen zu können und um festzustellen, wo die Fehlerursache ist.

Konfiguration & Dateipfade

Blender kann systemweit installiert oder aus einem extrahierten Archiv, welches alle benötigten Dateien enthält, gestartet werden.

Es gibt 3 verschiedene Ordner, welche Blender verwenden wird. Deren exakte Lokation ist betriebssystemabhängig.

- **LOCAL:** Lokation der Konfiguration und Laufzeitdateien (für enthaltene archive)
- **USER:** Lokation der Konfigurationsdateien (normalerweise das Home-Verzeichnis des Benutzers)
- **SYSTEM:** Lokation der Laufzeitdateien für die systemweite Installation (evtl. nur-lesend).

Für Systeminstallationen die Verzeichnisse **SYSTEM & USER** benötigt.

Für ein lokal extrahiertes Blender werden die Benutzerkonfiguration und Laufzeitdaten im selben Unterverzeichnis gehalten, was es erlaubt mehrere Instanzen von Blender ohne Konflikte zugleich zu starten, wobei die **USER** und **SYSTEM** Dateien ignoriert werden.

Im Folgenden die Standardlokationen für jedes System:

OSX

LOCAL: ./2.63/

USER: /Users/{user}/Library/Application Support/Blender/2.63/

SYSTEM: /Library/Application Support/Blender/2.63/

zu beachten ist, dass OSX die Blender Binärdaten im Ordner .blender.app/Contents/MacOS/blender abspeichert, somit ist der lokale Pfad zu Dateien und Konfigurationen .blender.app/Contents/MacOS/2.63

Windows

LOCAL: .\2.63\

USER: C:\Documents and Settings\{username}\AppData\Roaming\Blender Foundation\Blender\2.63\

SYSTEM: C:\Documents and Settings\All Users\AppData\Roaming\Blender Foundation\Blender\2.63\

Unix (Linux/BSD/Solaris)

LOCAL: ./2.63/

USER: \$HOME/.blender/2.63/

SYSTEM: /usr/share/blender/2.63/

zu beachten ist, dass ./2.63/ relativ zu Blenders ausführbaren Dateien ist und für enthaltene Archive, welche über die offizielle Seite blender.org verteilt werden, verwendet wird.

Pfad Layout

Im Folgenden wird das Pfadlayout beschrieben, welches für die zuvor aufgeführten Verzeichnisse verwendet wird.

Wobei beispielsweise ./config/**startup.blend** ~/.blender/2.63/config/startup.blend sein kann.

- ./autosave/ ...
Lokation der automatisch gespeicherten .blend-Dateien. *gilt nur für Windows, andere Systeme verwenden das temp-Verzeichnis..*
Suchreihenfolge: **LOCAL, USER**
- ./config/ ...
Standards und Sitzungsinformationen
Suchreihenfolge: **LOCAL, USER**
- ./config/**startup.blend**
standardmäßige Datei, welches beim Start geladen wird.
- ./config/**bookmarks.txt**
Lesezeichen im Dateibrowser.
- ./config/**recent-files.txt**
Liste der zuletzt verwendeten Dateien.
- ./datafiles/ ...
Laufzeitdateien
Suchreihenfolge: **LOCAL, USER, SYSTEM**
- ./datafiles/locale/{language}/
Übersetzungen. *wird derzeit nicht verwendet!*
- ./datafiles/icons/*.png
Icon Themen für Blenders Benutzerschnittstelle (UI). *derzeit nicht in den Themeneinstellungen wählbar.*
- ./datafiles/brushicons/*.png
Bilder für jeden Pinsel.
- ./scripts/ ...
Python Skripten für das UI und Werkzeuge
Suchreihenfolge: **LOCAL, USER, SYSTEM**
- ./scripts/addons/*.py

Python Erweiterungen, welche in den Benutzereinstellungen aktiviert werden können. Enthält die Unterstützung für Dateiimport/-export, Render Engine Integration und viele handliche Hilfsmittel.

- `./scripts/addons/modules/*.py`
Module, die für Erweiterungen verwendet werden können (wurde zu Pythons `sys.path` hinzugefügt)
- `./scripts/addons_contrib/*.py`
ein weiteres Verzeichnis für Erweiterungen, welches für gemeinschaftlich gewartete Erweiterungen bestimmt ist (muss manuell angelegt werden)
- `./scripts/addons_contrib/modules/*.py`
Module die für verteilte Erweiterungen verwendet werden können (wurde zu Pythons `sys.path` hinzugefügt)
- `./scripts/modules/*.py`
Python Module, die unsere Kern-API und Hilfsfunktionen für das Importieren anderer Skripten enthält (wurde zu Pythons `sys.path` hinzugefügt)
- `./scripts/startup/*.py`
Skripten, die beim Programmstart automatisch importiert werden.
- `./scripts/presets/{preset}/*.py`
Voreinstellungen, die für das Speichern von benutzerdefinierten Einstellungen für den Kleidung-Effekt, Render Formate usw. verwendet wird
- `./scripts/templates/*.py`
Beispielskripten, die über den Kopf des Textrahmens -> Text -> Skript Textschablonen erreichbar sind.
- `./python/ ...`
gebündelte Python Distribution, welche nur benötigt wird, wenn das Python des Systems fehlt oder inkompatibel ist.
Suchreihenfolge: **LOCAL**, **SYSTEM**

Anmerkungen

Pfad für Benutzerskripten

Der Pfad für benutzerdefinierte Skripten bietet eine Möglichkeit ein eigenes Verzeichnis zu definieren, welches sowohl für Skripten, als auch als Benutzerskripten Pfad verwendet wird. Dazu müssen Unterverzeichnisse, wie `startup/`, `addons/`, `modules/`, usw., analog zur Struktur Blenders Skriptenverzeichnis angelegt werden, da direkt kopierte Skripten in dieses Verzeichnis beim Start nicht geladen werden.

Umgebungsvariablen

Umgebungsvariablen können für das Überschreiben der Standardpfade verwendet werden, bzw: `$BLENDER_USER_CONFIG`, `$BLENDER_SYSTEM_PYTHON`.

Dieses Vorhaben benötigt normalerweise keine Einstellungen, kann jedoch nützlich für eigene Konfigurationen sein.

Weitere Details können im Abschnitt 'Environment Variables' in 'blender --help' eingesehen werden (englisch).

Skriptenpfade und fehlende Schaltflächen

Wenn Blender ohne UI startet, liegt es eventuell daran, dass die Skripten nicht korrekt geladen wurden. Die Ursachen können sein...

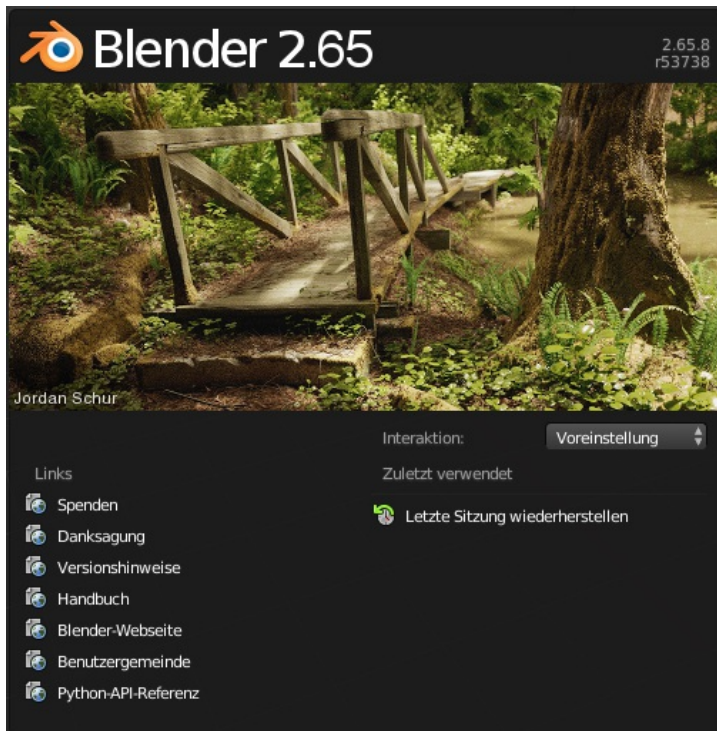
- der Skriptenpfad wurde nicht gefunden.
- ein Fehler in einem Skript.
- ein Versionskonflikt zwischen Blender und den Skripten.

Am Besten startet man Blender in diesem Fall aus einem Terminal/Befehlsfenster, um alle Fehlermeldungen sehen zu können und um festzustellen, wo die Fehlerursache ist.

Der erste Start mit Blender

Wenn Sie bereits den Umgang mit Blender 2.4x oder anderer 3D-Software, wie Maya, 3ds Max oder XSI kennen, so werden Sie sofort bemerken, dass Blender 2.5 und 2.6 eindeutig anders aussieht, als Sie es gewohnt sind. Dennoch werden Sie schon nach kurzer Zeit Gemeinsamkeiten mit herkömmlicher 3D-Software, wie beispielsweise den "3D-Viewport", den "Outliner" oder die "Timeline" entdecken. Wenn dies aber Ihre erste Benutzung einer 3D-Software ist, so werden Sie wahrscheinlich leicht verwirrt sein. Glücklicherweise gibt es nur eine einzige Regel, was das Lernen von 3D mit Blender angeht: Fürchten Sie sich nicht vor neuen Erkundungen und Experimenten!

Nachdem Sie Blender gestartet haben, schauen Sie kurz in die rechte obere Ecke ihres Startbildschirms. Dort steht, welche Version von Blender Sie nutzen.



Die linke Seite zeigt Ihnen einige nützliche Links, wie z.B. den "[release log](#)" der Version, die Sie nutzen (Neuerungen dieser Version), das Wiki-Handbuch/ [Manual](#) (was sie jetzt gerade lesen) und die offizielle [Blender Website](#). Diese Links finden Sie ebenfalls über das Help-Menü.

Die rechte Seite listet die zuletzt genutzten Blender-Dateien (.blend) auf. Wenn Sie Blender das erste Mal starten, wird dieser Bereich leer sein. Diese Liste ist auch unter File » Open Recent verfügbar. Über das "Interaction"-Menü können Sie Tastatur-Voreinstellungen wählen (Blender or Maya).

Um mit dem Nutzen von Blender zu beginnen, haben Sie drei Optionen:

- Klicken Sie auf eine der zuletzt benutzten Dateien (vorausgesetzt Sie haben welche)
- Klicken Sie auf irgendeine andere Fläche des Programmes, ausgenommen des Startbildschirms oder
- Drücken Sie Esc um ein neues Projekt zu starten

Speichern Sie ihre Arbeit regelmäßig

Blender warnt Sie nicht vor ungespeicherten Daten, wenn Sie das Programm schließen. Allerdings sind nicht alle ihre Daten verloren, wenn Sie Blender ohne vorherigem Speichern beenden. Öffnen Sie Blender einfach wieder und drücken Sie auf Recover Last Session im Startbildschirm. Diese Option steht Ihnen auch unter File » Recover Last Session zur Verfügung.

Temporäre .blend Datei

Jedes Mal, wenn Blender geschlossen wird, speichert es die aktuellen Daten in einer temporären .blend Datei ab. Wenn Sie ihre letzte Sitzung wiederherstellen, lädt Blender die Daten aus dieser Datei.

Konzepte der Benutzeroberfläche



Blender wurde als eine plattformübergreifende Software entwickelt. Sie läuft also sowohl unter Linux, Mac OS X und Windows. Der Grund hierfür ist, dass Blender auf [OpenGL](#) basiert, welche im Einklang mit allen drei Betriebssystemen steht.

Die 3 Regeln

Die Blender Benutzeroberfläche baut auf 3 grundlegenden Prinzipien auf:

- **Keine Überlappung:** Das Interface ermöglicht es Ihnen, alle relevanten Werkzeuge und Optionen übersichtlich auf einen Blick zu sehen. Sie müssen keine Fenster herum schieben, welche sich überlappen könnten.⁽¹⁾
- **Keine Blockierungen:** Werkzeug- und Interface-Optionen blockieren den Benutzer nicht, andere Teile Blender zu nutzen. Blender öffnet auch keine Pop-Up-Fenster, welche Sie erst ausfüllen müssen, bevor die eigentliche Anwendung vollzogen wird.
- **Keine Modalität:** Benutzereingaben sollten so einheitlich und berechenbar wie möglich sein, ohne herkömmliche Methoden zu verändern (Maus, Tastatur).

⁽¹⁾Allerdings erlaubt Blender 2.6 dennoch mehrere Fenster für die Verwendung unter mehreren Bildschirmen. Es ist eine Ausnahme der *Keine-Überlappungen-Regel*.

Leistungsfähige Benutzeroberfläche



Blenders Benutzeroberfläche wurde komplett in [OpenGL](#) entwickelt, welches erlaubt, die Oberfläche an sämtliche Bedürfnisse anzupassen. Fenster und andere Elemente der Benutzeroberfläche können geschwenkt, gezoomt und deren Inhalte bewegt werden. Sie können Ihren Bildschirm ganz nach Ihrem Geschmack für jede spezielle Aufgabe anpassen und diese Einstellungen auch abspeichern.

Blender benutzt zudem viele Tastatur-Befehle, um die Arbeitsgeschwindigkeit weiter zu erhöhen. Diese können nach Belieben geändert werden, so dass Sie sich diese noch besser merken können.

Überblick

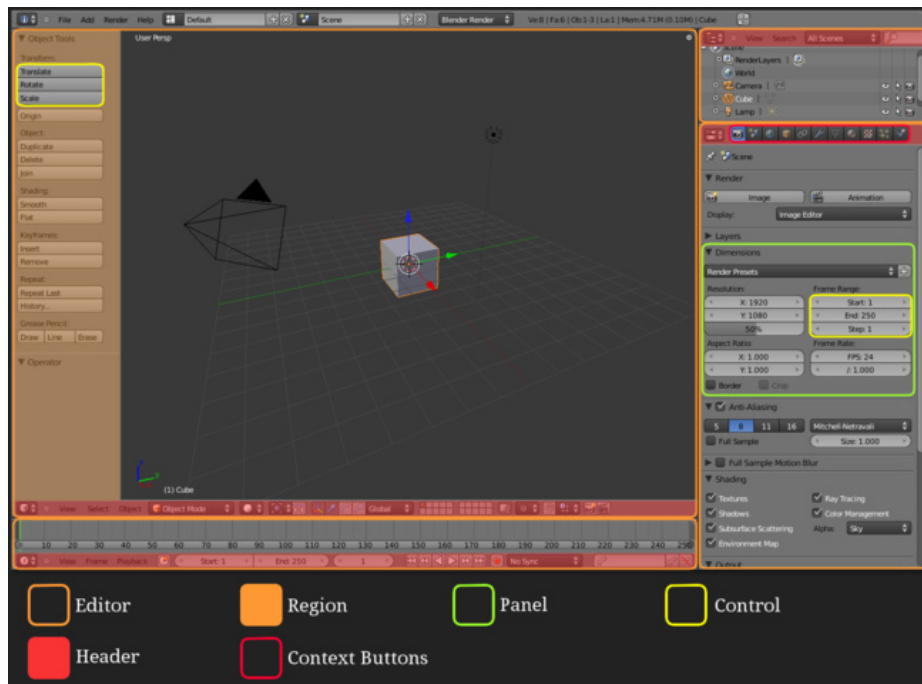
Lasst uns einen Blick auf die Standard-Oberfläche von Blender werden: Es besteht aus Editors, Headers, Context buttons, Panel, und Controls.

- In Blender nennen wir den **Editor** den Teil der Software, welche eine ganz bestimmte Funktion hat (3D view, Properties Editor, Video Sequence Editor, Nodes Editor...). Jeder Editor hat seinen eigenen *Header* am oberen oder unteren Rand.
- **Context buttons** erlauben den Zugriff auf Optionen. Sie ähneln Tabs und sind oftmals auf einem Header eines Editors platziert.
- Auf jedem Editor sind die Optionen in **Panel** gruppiert, um das Interface logisch zu organisieren. (Shadow panel, Color panel, Dimensions panel...).
- **Sidebars** sind nur in einigen Editoren enthalten. In solchen Fällen sind dort die Panel und Controls gruppiert. Um den Arbeitsbereich zu optimieren, ist es möglich Sidebars temporär zu verstecken.
- Panel beinhalten **Controls**. Mit Controls können Sie Einfluss auf Funktionen, Optionen oder Werte nehmen. In Blender gibt es verschiedene Arten von Controls :
 - **Buttons:** Erlauben den Zugriff auf ein Werkzeug (Verschieben, Drehen). Diese Werkzeuge haben normalerweise Tastaturbefehle um Ihre Arbeitsgeschwindigkeit zu erhöhen. Um den jeweiligen Tastaturbefehl sichtbar zu machen, halten Sie einfach Ihre Maus über einen solchen Button.
 - **Checkboxes:** Ermöglichen das Aktivieren bzw. Deaktivieren einer Option. Solch ein Control kann nur einen Boolean-Wert besitzen (Wahr/Falsch, 1/0).
 - **Sliders:** Erlauben das Eingeben von Float-Werten. Diese können limitiert sein (von 0.0 bis 100.0) oder auch nicht (von -∞ bis +∞). Es existieren also zwei Arten von Slidern in Blender.

-

Select menus: Ermöglichen das Auswählen eines Wertes aus einer Liste. Der Unterschied zu einer Checkbox ist, dass Werte hier benannt sind und es auch mehr als nur zwei Werte gibt.

[Lesen Sie mehr über Knöpfe und Steuerelemente »](#)



Eingabegeräte

Blender's Benutzeroberfläche ist entworfen worden, um mit den folgenden empfohlenen Eingabegeräten verwendet zu werden:






- Eine 3-Tasten-Maus inklusive Mausrad
- Eine herkömmliche Tastatur inklusive NumPad
- NumLock sollte generell an sein

Wenn Sie nicht die empfohlenen Eingabegeräte besitzen (z.B. Nutzung eines Laptops), ist es über die *Benutzereinstellungen* möglich, diese Konfiguration zu emulieren.

[Lesen Sie mehr über die Blender-Konfiguration](#)

Veranschaulichungen in diesem Handbuch

Dieses Handbuch benutzt die folgenden Veranschaulichungen um die Benutzereingaben darzustellen:

- Maustasten:
 - LMB  - Linke Maustaste
 - MMB  - Mittlere Maustaste
 - RMB  - Rechte Maustaste
- Mausrad
 - MMB  - bezieht sich auf das Klicken des Mausrads als Mittlere Maustaste
 - Wheel  - Drehen des Mausrads
- Tastaturbefehle werden in diesem Handbuch wie auf einer Tastatur dargestellt:
 - G - bezieht sich auf das kleine "g".
 - ⇧ Shift, Ctrl und Alt sind generell als Modifikator-Tasten festgelegt
 - CtrlW oder ⇧ ShiftAltA - zeigt an, dass diese Tasten gleichzeitig gedrückt werden müssen
 - 0 NumPad bis 9 NumPad, + NumPad - usw. beziehen sich auf die Tasten des NumPads

Andere Tasten werden mit ihrem Namen angegeben, wie z. B. Esc, ⇐ Tab, F1 bis F12. Besonders hervorzuheben sind die Pfeiltasten: ←, → usw.

Allgemeine Nutzung

Da Blender eine umfassende Nutzung von Maus und Tastatur vorsieht, gibt es eine goldene Regel, welche sich unter Blender-Anwendern entwickelt hat: **Legen Sie eine Hand auf die Maus und die andere auf die Tastatur.** Die am häufigsten verwendeten Tasten sind so gruppiert, dass sie mit der linken Hand in Standardposition (Zeigefinger auf F) bei einem englischen Layout mühelos erreicht werden können. Demzufolge legen Sie also ihre rechte Hand auf die Maus.










Wenn Sie normalerweise eine Tastatur benutzen, welche sich deutlich vom englischen Tastatur-Layout unterscheidet, sollten sie über eine Layout-Änderung nachdenken. Natürlich können Sie auch einfach sämtliche Tastaturbefehle Ihrem Layout anpassen, allerdings basiert dieses Handbuch auf den Standardeinstellungen.



[Lesen Sie mehr über die Blender-Konfiguration](#)

Maustasten-Emulation

Wenn Sie keine 3-Tasten-Maus verwenden, müssen Sie eine solche emulieren. Dies können Sie über die [Benutzer Voreinstellungen](#) tun.

Die folgende Tabelle zeigt Ihnen die benutzten Kombinationen:

3-Tasten-Maus	2-Tasten-Maus	Apple-Maus
LMB 	LMB 	LMB  (Maustaste)
MMB 	Alt LMB 	⇧ Opt LMB  (Option/Alt-Taste + Maustaste)
RMB 	RMB 	⇧ Cmd LMB  (Befehls-Taste/Apfel-Taste + Maustaste)

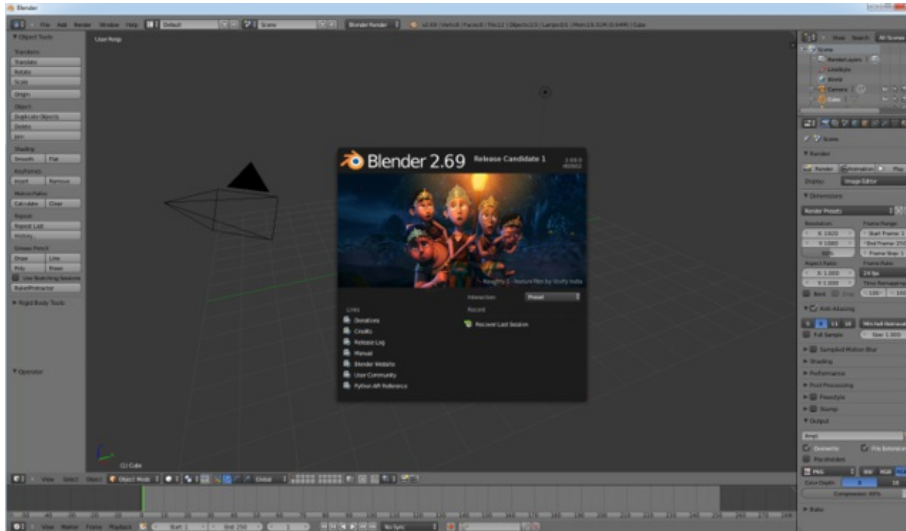
Alle Maus/Tastatur-Kombinationen, welche in diesem Handbuch vorkommen, können mit den in der Tabelle aufgelisteten Kombinationen ausgedrückt werden. Zum Beispiel wird ⇧ ShiftAlt RMB  zu ⇧ ShiftAlt⇧ Cmd LMB  bei einer Apple-Maus.

NumPad Emulation

[Lesen Sie mehr über die NumPad Emulation auf der Benutzer-Voreinstellungen-Seite](#)

Das Fenstersystem

Beim Starten von Blender sollte das folgende Bild zu sehen sein (der Begrüßungsbildschirm in der Mitte wird mit neueren Versionen wechseln):

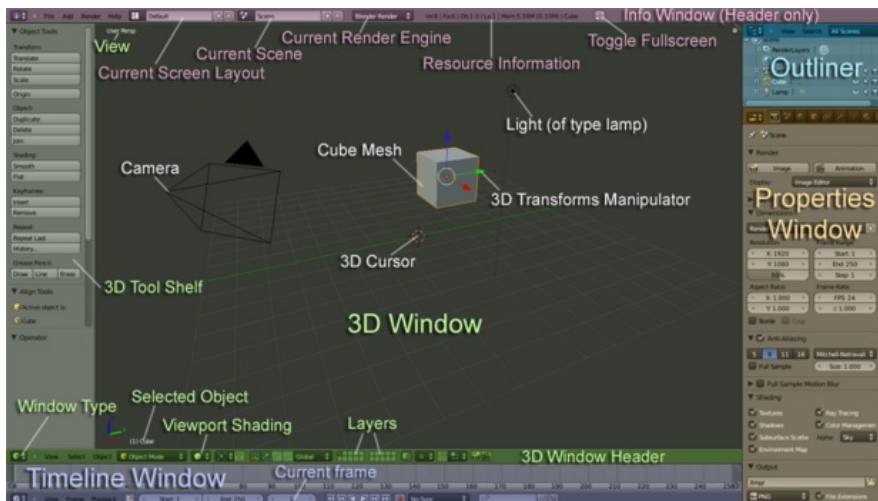


In der Mitte des Fensters befindet sich der Begrüßungsbildschirm. Dieser bietet einen schnellen und einfachen Einstieg in zuletzt geöffnete Blender-Dateien. Zum Starten eines neuen Projektes führt ein Klick neben das Bild dazu, dass dieses verschwindet und das Standardlayout und ein Würfel – also die normale Startkonfiguration – zurückbleibt.

Jedes sichtbare Fenster kann weiterhin in weitere Segmente unterteilt werden (beschrieben im jeweiligen Abschnitt in [Ausrichtung von Rahmen](#)). Die Standardkonfiguration wird im Folgenden beschrieben.

Die Standardkonfiguration

Die Standardkonfiguration wird jedes mal geladen wenn Blender gestartet oder eine neue Datei angelegt wird.



The default blender scene

Diese ist in fünf Fenster unterteilt:

- Das Informationsfenster und Hauptmenü (Info window) ganz oben, welches nur den Kopf zeigt (da dort nichts weiter ist).
- Ein großes 3D Fenster (3D window)
- Ein Zeitleistenfenster unten (Timeline window)
- Eine Inhaltsübersicht (Outliner) oben rechts.
- Ein Einstellungenfenster (Properties window) unten rechts.

Als Einführung werden hier einige fundamentale Elemente abgedeckt.

Das Informationsfenster und Hauptmenü


Siehe oben gezeigtes Bild als Referenz.

- **Derzeitige Ansicht (Standard ist Default):** Standardmäßig bietet Blender einige vorkonfigurierte Screens (Sichten) aus denen gewählt werden kann. Sofern benötigt können weitere [individuelle Sichten](#) kreiert und benannt werden.
- **Derzeitige Szene:** Mehrere [Szenen](#) erlauben es an mehreren virtuellen Umgebungen mit komplett getrennter Datenhaltung oder miteinander verlinkten Objekten und/oder Maschendaten zu arbeiten. (Einige 3D Anwendungen erstellen Szenen in eigenen Dateien, während Blender in einer .blend Datei mehrere Szenen speichern kann).

- **Derzeitige Rendering Engine:** Bietet eine Liste verfügbarer Rendering Engines.
- **Ressourcen Informationen:** Bietet Informationen über Blender und benutzte Systemressourcen. Dieser Abschnitt bietet sowohl eine Übersicht über die verwendete Speichermenge gegenüber der Anzahl an Eckpunkten (vertices), Flächen (faces) und Objekten (objects) der selektierten Szene, als auch die Summe der derzeit verwendeten Ressourcen. Diese Hilfe zeigt, wann das System/die Hardware an die Grenzen stößt.
- **Vollbild (Toggle Fullscreen):** Ein- und Ausschalten des Vollbildmodus.

3D Ansicht (3D Window)

Siehe oben gezeigtes Bild als Referenz.

- **Fenstertyp (Window Type):** Erlaubt das Ändern der Art des Fensters (was auch in jedem Fensterkopf zu sehen ist). Wenn zum Beispiel die Inhaltsübersicht (Outliner) zu sehen sein soll kann dies hier gewählt werden. Weitere Informationen dazu: [Komplette Liste von Fenstertypen](#)
- **3D Transformierer (3D Transform Manipulator):** Dieser ist eine visuelle Hilfe zum transformieren von Objekten (greifen/bewegen (G), rotieren (R) und skalieren (S)). Objekte können auch mithilfe von Tastaturkürzeln transformiert werden: (G/R/S); StrgLeertaste verändert die Sichtbarkeit des Manipulators, was auch per Klick auf das Koordinatensystem-Symbol auf der Werkzeugeiste möglich ist. Die Translations-/Rotations-/Skalierungsmanipulatoren können per Klick auf jedes der drei Symbole rechts des Koordinatensystem-Symbols gewählt werden. Das Klicken der linken Maustaste (⇧ Shift LMB ) auf eines der Symbole wird den jeweiligen Manipulator hinzufügen/entfernen.



- **3D Zeiger (3D Cursor):** Dieser kann mehrere Funktionen beinhalten. Beispielsweise repräsentiert er, wo neue Objekte beim erstmaligen kreieren platziert werden, oder wo sich der Mittelpunkt einer Rotation befindet. Hier ein Bild eines 3D Zeigers, isoliert vom Rest der Szene:



- **Würfel Masche (Cube Mesh):** Standardmäßig wird eine Neuinstallation von Blender immer mit einer Würfel Masche (Cube Mesh) starten, welche im Zentrum des gesamten 3D Raumes sitzt (im obigen Bild wurde sie bereits verschoben). Nach einiger Zeit wird es wahrscheinlich nötig die Standardkonfiguration zu verändern; Dies wird mit dem [Konfigurieren von Blender](#) realisiert. Eine Standardkonfiguration wird als solche mit dem Tastaturkürzel StrgU gespeichert (Speichern von Standardeinstellungen).
- **Licht (vom Typ Lampe):** Standardmäßig wird eine Neuinstallation von Blender immer mit einer Lampe Light starten, welche irgendwo nahe des globalen Mittelpunktes des 3D Raumes positioniert ist.
- **Kamera (Camera):** Standardmäßig wird eine Neuinstallation von Blender immer mit einer Kamera Camera starten, die irgendwo nahe des globalen Mittelpunktes des 3D Raumes positioniert ist und auf diesen gerichtet ist.
- **Selektiertes Objekt (selected object):** Dieses Feld zeigt den Namen des derzeit gewählten Objektes.

3D Fenster Kopf (3D Window Header)

Dies ist der Kopf für das 3D Fenster. Alle Fenster in Blender haben einen Kopf (auch wenn dieser Kopf am Fuß erscheint, wird er dennoch Kopf genannt)

Mehr unter [Blender headers](#) »

Siehe oben gezeigtes Bild als Referenz.

- **Form der Darstellung (Viewport shading):** Blender rendert das 3D Fenster mithilfe von [OpenGL](#). Die Form der Darstellung kann hier per Klick auf diese Schaltfläche und Selektion der gewünschten Form gewählt werden. Die Auswahl startet bei Boxen bis hin zu komplexen beschatteten Texturen. Für den texturierten Stil wird eine rechenstarke Grafikkarte empfohlen.
- **Ebenen (Layers):** Ebenen erleichtern Modellieren und Animieren. Diese sind eine Hilfestellung für das Ordnen von Objekten in funktionale Gruppen. Beispielsweise enthält eine Ebene ein Wasser Objekt und eine andere Bäume oder eine Ebene enthält Kameras und Lichter. Zum entrümpeln der Sicht können Ebenen ein- und ausgeschaltet werden.

Einstellungen Fenster (Properties Window)

Der Kopf des Einstellungen Fensters ist ein wenig dunkler gefärbt, als im obigen Bild. Dieses gilt als Referenz zur folgenden Beschreibung.

Das Schaltflächen (Einstellungen) Fenster zeigt gruppierte Leisten. Im Kopf des Fensters befindet sich eine Reihe von Schaltflächen (sogenannte Kontextboxen), welche die Möglichkeit bieten die verschiedenen Gruppen von Leisten zu zeigen.




Die Leisten im Fenster selbst helfen beim Gruppieren ähnlicher Schaltflächen und Bedienelemente. Einige Leisten sind sichtbar oder unsichtbar, abhängig vom Typ des gewählten Objektes. Leisten können mithilfe des kleinen Pfeils links des Leistentitels (z.B. neben *Render*) eingefahren und per ziehen der obigen rechten Ecke neu angeordnet werden.

Inhaltsübersicht (Outliner Window)

Dieses Fenster listet alle Objekte einer Szene auf. Es ist beim Arbeiten mit großen Szenen mit vielen Elementen sehr hilfreich.

Im Kopf des Fensters kann nach Elementtypen gefiltert und deren Darstellung bestimmt werden.

Zeitleiste (Timeline Window)

Dieses Fenster bietet eine Zeitleiste durch die mithilfe der linken Maustaste (LMB ) gefahren werden kann.

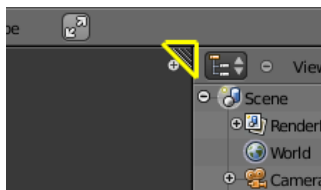
Organisieren von Frames (Rahmen)

Blender verwendet einen neuartigen Ansatz zur Bildschirmteilung, um Frames zu organisieren. Das Anwendungsfenster ist immer ein Rechteck auf der Arbeitsfläche. Blender unterteilt dieses in einige in der Größe veränderbare Fensterframes. Ein Fensterframe enthält den Arbeitsbereich eines bestimmten Fenstertyps, wie beispielsweise eine 3D-Ansicht oder eine Inhaltsübersicht. Die Idee dahinter ist die Unterteilung eines großen Anwendungsfensters in jegliche Anzahl kleinerer (jedoch immer noch rechteckiger), nicht überschneidender Fensterframes. Somit ist jedes Fenster stets vollständig sichtbar und es ist sehr einfach von einem Fenster in ein anderes zu wechseln, um dort weiter zu arbeiten.

Ein Fenster maximieren

Ein Fensterframe kann mithilfe des View → Toggle Full Screen Menüeintrags maximiert werden, um das komplette Anwendungsfenster zu füllen. Um zur normalen Größe zurückzukehren betätigt man die Schaltfläche erneut. Ein schneller Weg um dies zu erreichen ist das Verwenden der Tastaturkürzel ⇧ ShiftLeertaste, Strg↓ oder Strg↑, um zwischen maximierten und gerahmten Fenstern hin- und her zu wechseln. BEACHTET: Das Fenster, über dem aktuell der Mauszeiger steht wird dasjenige sein, welches mithilfe der Tastaturkürzel maximiert werden wird.

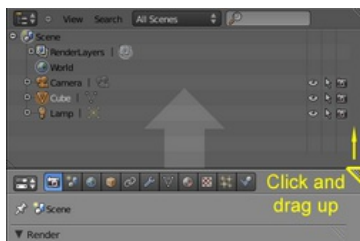
Unterteilen eines Fensters



In den Ecken oben rechts und unten links eines Fensters befinden sich die Unterteilungsvorrichtungen, welche wie geriffelte Haltegriffe aussehen. Beides unterteilt und kombiniert Fensterausschnitte. Beim Überfahren wechselt der Mauszeiger zu einem Kreuz. Mit Klicken und ziehen der linken Maustaste (LMB) nach links oder nach unten wird der Ausschnitt horizontal oder vertikal unterteilt.

Zwei Frames vereinen

Um zwei Fensterrahmen zu vereinigen müssen diese in Vereinigungsrichtung die selbe Dimension haben. Beispielsweise müssen zwei Fenster nebeneinander dieselbe Höhe haben. Wenn das linke nicht die selbe Höhe wie das rechte hat, wird es nicht möglich sein beide horizontal zu vereinen. Das liegt daran, dass das kombinierte Fenster wieder ein Rechteck ergeben muss. Die selben Regeln gelten für zwei Fensterrahmen übereinander, beide müssen die selbe Breite haben. Falls das obere horizontal geteilt ist muss es zunächst zusammengefügt werden. Dann kann das untere mit dem oberen vereint werden.



Um das aktuelle Fenster mit dem darüber zu vereinigen (im Bild wird das Eigenschaften-Fenster "über" die Inhaltsübersicht vereinigt) fährt man zunächst mit dem Mauszeiger über den Fensterleiter. Wenn der Zeiger zu einem Kreuz wird führt Klicken und Ziehen mit LMB zum Vereinigungsprozess. Das obere Fenster wird ein wenig dunkler erscheinen und mit einem nach oben zeigenden Pfeil überlagert sein. Dies zeigt, dass beim Loslassen der linken Maustaste der untere (aktuelle) Rahmen den Raum des dunkleren Rahmen mit "übernehmen" wird. Ein Bewegen des Mauszeigers zurück in den unteren Bereich sorgt dafür, dass dieser vom oberen vereinnahmt wird.

Ebenso werden Fenster von links nach rechts oder umgekehrt vereint.

Abgebrochen wird die Operation mit Drücken der Esc-Taste noch vor Loslassen der linken Maustaste.

Verändern der Fenstergröße

Die Fenstergröße kann verändert werden, indem dessen Grenzen mithilfe der LMB gezogen werden. Dazu lediglich über die Grenze zweier Fenster fahren, bis der Mauszeiger zu einem nach zwei Seiten zeigenden Pfeil wird und klicken und ziehen.

Inhalte tauschen

Inhalte zweier Rahmen können mithilfe der Maus und Tastenkombination Strg LMB angewendet auf die Unterteilungsvorrichtung getauscht werden. Ein Ziehen des Mauszeigers in den Zielrahmen tauscht die Inhalte. Die Rahmen müssen sich dazu nicht direkt nebeneinander, jedoch im selben Fenster befinden.

Neue Fenster öffnen

Möglicherweise ist es nötig neue Vollbildfenster mit Blender Rahmen zu öffnen. Dies kann beispielsweise bei mehreren Monitoren, die verschiedene Informationen der selben Blender Instanz zeigen, hilfreich sein.

Alles, was dazu nötig ist, ist die Maus und Tastenkombination ⇧ Shift LMB auf einen Rahmenteiler. Bewegen des Zeigers löst damit ein neues Fenster heraus. Dieses stellt eine Kopie des herausgezogenen Rahmens dar.

Das neue Fenster einmal erstellt, kann es auf einen anderen Monitor gezogen werden (oder verbleiben). Dessen Größe kann verändert werden und analog zu allen anderen Rahmen unterteilt, zusammengefügt, usw. werden.

Ein anderer Weg ein separates Fenster zu erhalten geht über *File* → *User Preferences...* (oder StrgAltU). Dies öffnet die Benutzereigenschaften mit dem analog zum Fenster im vorherigen Absatz verfahren werden kann.

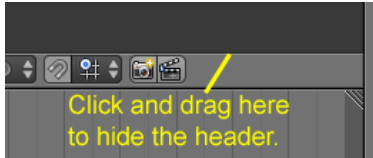
Header (Menüleisten, Werkzeugleisten)

Alle Fenster haben einen Header (Kopf) (der Streifen mit hellerem, grauen Hintergrund, welcher Schaltflächen mit Piktogrammen enthält). Im Folgenden wird der Header auch *Menüleiste* oder *Werkzeugleiste* des Fensters genannt. Der Header befindet sich entweder oben (wie im Eigenschaften Fenster) oder unten (wie im 3D Fenster) im Fensterbereich. Das folgende Bild zeigt den Header des 3D-Ansichtsfensters:



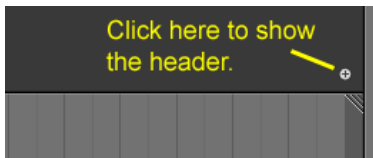
Beim Überfahren mit dem Mauszeiger ändert sich die Headerfarbe in ein helleres Grau. Dies bedeutet, dass es fokussiert ist. Alle Tastaturkürzel betreffen nun die Inhalte dieses Fensters.

Einen Header verstecken



Der Mauszeiger muss sich über der dünnen Linie zwischen einem Fenster und dessen Header befinden, um einen Header zu verstecken. Der Zeiger nimmt dann die Form eines Auf-Ab-Pfeils an. Nun LMB gedrückt halten und über den Header ziehen, um ihn zu verstecken.

Einen Header sichtbar machen



Ein versteckter Header hinterlässt ein kleines Plus-Zeichen (siehe Bild). Ein Klick auf dieses lässt den Header wieder erscheinen.

Anmerkung 1: Im 3D Fenster befinden sich dort bis zu zwei zusätzliche Plus-Zeichen (im Fenster oben links und rechts). Diese öffnen Steuerleisten mit einigen Werkzeugen, nicht einen zweiten Header.



Anmerkung 2: In manchen Fenstern kann das erwähnte Plus-Zeichen schwer zu finden sein, da es wie ein Teil anderer Symbole aussehen könnte. Ein Beispiel ist die Inhaltsübersicht, in der sich weitere derartige Plus-Symbole befinden. Daher hat der Hintergrund des Plus-Zeichens, das den Header zurück Holt eine Hintergrundfarbe erhalten, die sich abhebt.

Header-Position

Ein RMB-Klick auf den Header und Wählen der entsprechenden Option im erscheinenden Menü bewegt einen Header von oben nach unten oder umgekehrt. Befindet sich der Header oben, so trägt die Option den Namen "Flip to Bottom". Befindet er sich unten, heißt sie "Flip to Top".



Thema Farben

Blender erlaubt Farbeinstellungen für die meisten Komponenten, um den Wünschen des Benutzers gerecht zu werden. Falls die Farben der Bildschirmfotos nicht mit denen der eigenen Installation übereinstimmen, kann es daran liegen, dass das Standard-Thema geändert wurde. Das Erstellen eines neuen Themas oder Wählen/Ändern eines existierenden Themas geschieht über das [Benutzereinstellungen](#)-Fenster und Klicken auf den Themes-Reiter des Fensters.

Fenstertyp-Schaltfläche

Mit Klicken des ersten Icons am linken Ende des Headers kann aus 16 verschiedenen Fenstertypen gewählt werden. Jeder Fensterrahmen in Blender wird möglicherweise jede Art eines Fensters enthalten. Sollen sich beispielsweise überall 3D-Ansichten befinden, kann dies über das Ändern aller Fenster in den entsprechenden Typ geschehen.

Menüs und Schaltflächen

Die meisten Fenster Header, die sich gleich neben der ersten "Fenster-Typ" Schaltfläche befinden, stellen eine Zusammenstellung aus Menüs aus, die versteckt werden können - wieder mit einem kleinen Minus-Zeichen. Falls also ein Menü, welches irgendwo erwähnt wurde, nicht aufzufinden ist, ist es eventuell verborgen und lässt sich per Klick auf das Plus-Symbol neben der Fenstertyp-Schaltfläche wieder anzeigen.

Menüs erlauben es, viele Fähigkeiten und Kommandos direkt zu erreichen. Sie warten nur darauf entdeckt zu werden. Alle Menüeinträge zeigen die relevanten Tastaturkürzel, sofern sich dort welche befinden.

Menüs und Schaltflächen werden sich mit dem gewählten Fenstertyp, dem selektierten Objekt und dem Modus ändern. Sie zeigen jeweils nur die Aktionen, die ausgeführt werden können.

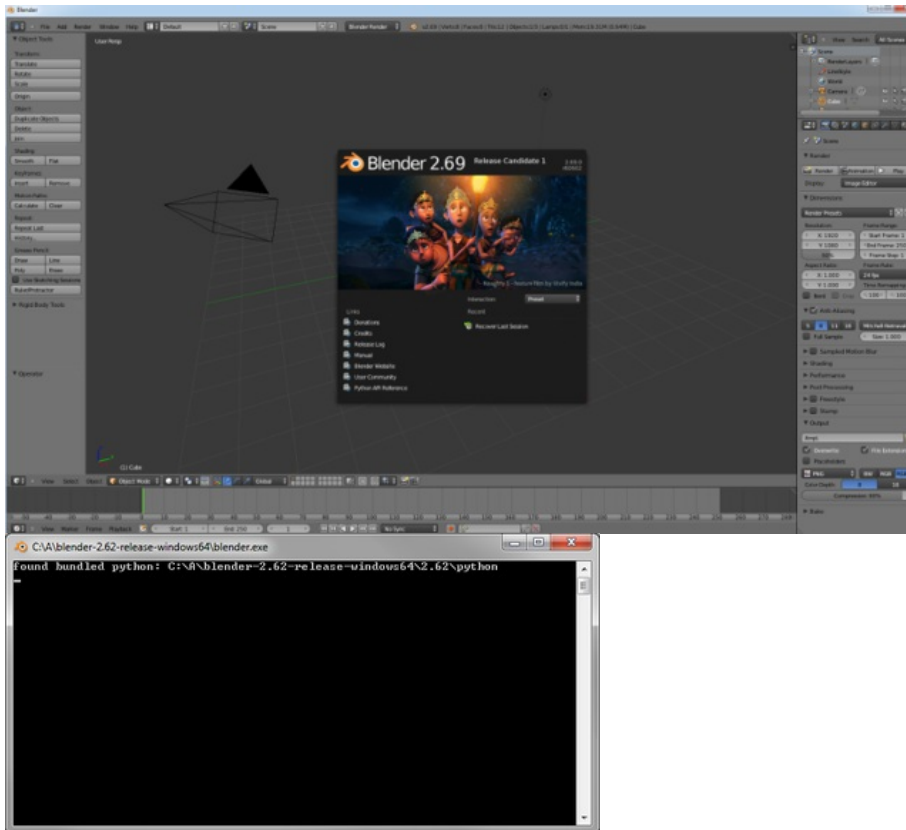
Das Konsolenfenster

Das Konsolenfenster ist ein Textfenster des Betriebssystems, welches Nachrichten über Operationen, Status und interne Fehler von Blender zeigt. Falls Blender abstürzt könnte das Konsolenfenster beim Identifizieren des Grundes helfen.

Windows XP/Vista/7

Wird Blender auf einem Windows Betriebssystem gestartet, so wird zunächst ein Konsolenfenster als ein separates Fenster auf dem Desktop kreiert. In Erwartung, dass die korrekten Konventionen zum Start zutreffend sind, sollte auch das Blender Hauptfenster erscheinen und das Konsolenfenster verschwinden. Dieses Vorgehen ist unterschiedlich zu dem der 2.4x Versionen, wo das Konsolenfenster stets im Hintergrund offen blieb, während das Hauptfenster im Vordergrund zu sehen war. Das Konsolenfenster in aktuellen Versionen lässt sich über Help » Toggle System Console anzeigen.

Der folgende Screenshot zeigt die zwei Fenster auf Windows 7.

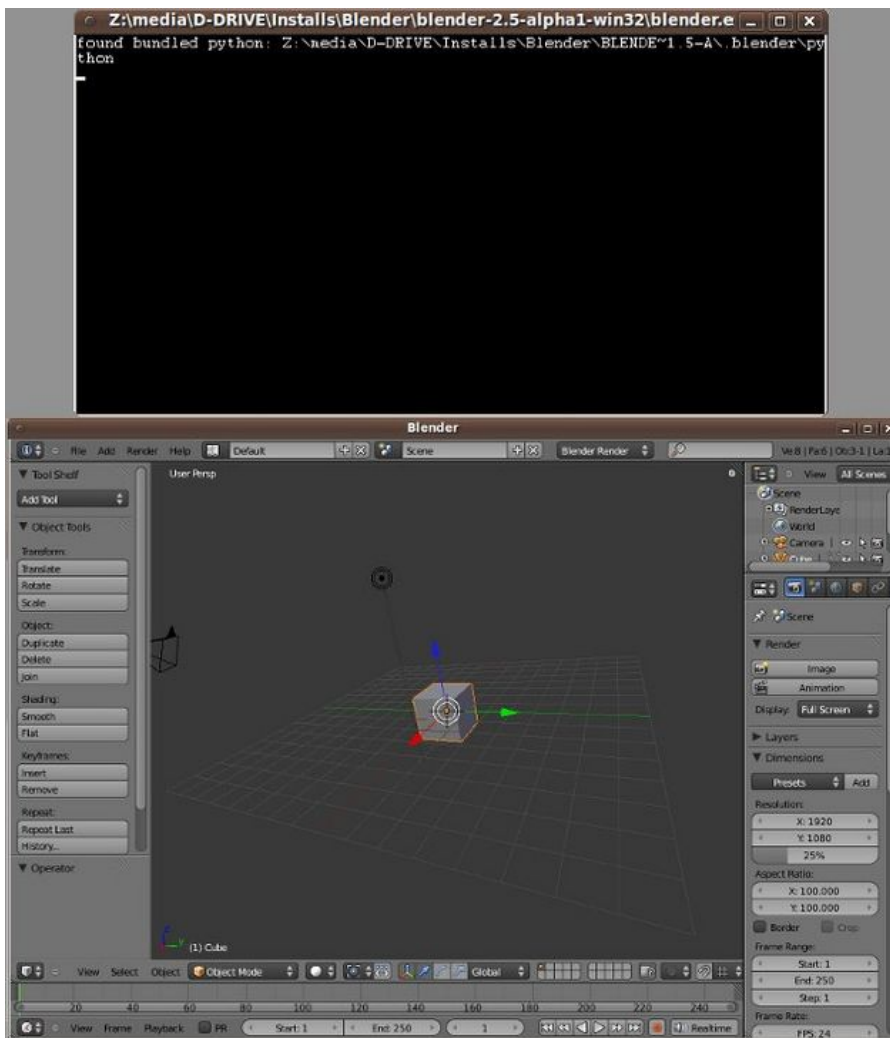


Linux

Das Blender Konsolenfenster in Linux wird generell nur sichtbar sein, wenn Blender von einem Terminal/Konsolenfenster aus gestartet wurde. Blender verwendet dann das Konsolenfenster aus dem es gestartet wurde.

Abhängig der Einstellungen der Desktopumgebung wird nach der Installation ein Blender-Icon auf dem Desktop oder ein neuer Blender Eintrag im Menü erscheinen. Falls Blender von einer dieser Varianten aus gestartet wird, werden die Blenderausgaben höchstwahrscheinlich auf dem Terminal geschrieben, auf dem der X-Server gestartet wurde ([X_Window_System](#)).

Dieser Screenshot zeigt Blender, gestartet von einem Linux Terminal/Konsolenfenster, und der resultierenden Konsole auf der die Ausgaben geschrieben werden.



Blender in Linux started from a Terminal



Das Blender Konsolenfenster schließen

Das Blender Konsolenfenster muss so lang offen gelassen werden, wie Blender verwendet wird. Das Schließen des Konsolenfensters wird auch Blender schließen und ungespeicherte Änderungen gehen verloren. Die MS-DOS Konsolenfenster und Blender Konsolenfenster können ähnlich aussehen. Es sollte also stets darauf geachtet werden, welches Fenster geschlossen wird.

MacOS

Der Prozess in MacOS ist dem von Linux sehr ähnlich, jedoch gibt es einen großen Unterschied. In Linux befindet sich Blender in einem Ordner, jedoch verwendet MacOS "Dateien" mit der Endung .app, welche "Anwendungen" genannt werden. Diese "Dateien" sind eigentlich Ordner, welche als Dateien im Finder erscheinen. Um also Blender zu starten muss im Terminal "cd blender.app" eingegeben werden (das Terminal befindet sich unter Anwendungen -> Werkzeuge), vorausgesetzt man befindet sich im Ordner, der die Applikation enthält. Von dort navigiert man zu den Unterverzeichnissen "Contents", "MacOS". Hier kann Blender mithilfe des Kommandos "./blender" gestartet werden, was dieselbe Vorgehensweise wie in Linux darstellt.

Konsolenfenster und Fehlermeldungen

Das Konsolenfenster kann viele verschiedene Typen von Status- und Fehlermeldungen anzeigen. Einige Meldungen informieren den Anwender lediglich über Dinge, die Blender tut. Diese Meldungen geben jedoch keine tatsächliche Auskunft über Blenders Funktionsfähigkeit. Andere Meldungen können relevante Fehler identifizieren, welche Blender höchstwahrscheinlich davon abhalten eine spezielle Aufgabe auszuführen und zum Stillstand zu bringen oder gänzlich abzustürzen. Die Meldungen des Blender Konsolenfensters können auch aus den Interna herrühren, wie beispielsweise [Python Skripten](#) und [Plugins](#)

Gewöhnliche Meldungen

- found bundled python: (FOLDER)

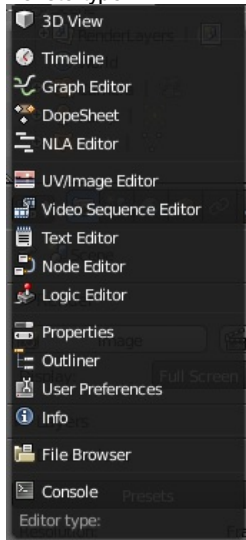
Die Meldung zeigt, dass es Blender möglich war die [Python](#) Bibliothek für den Python Interpreter zu finden, der in Blender eingebettet ist. Falls dieser Ordner fehlt oder nicht auffindbar ist, wird wahrscheinlich ein Fehler auftreten und diese Meldung nicht erscheinen.

- malloc returns nil()

Wenn Blender Operationen ausführt, welche zusätzlichen Speicher (RAM) erfordern, ruft es eine Funktion namens malloc (kurz

für Speicherallokierung) auf, welche versucht, die erwünschte Menge an Speicher für Blender zu allokkieren. Falls dies nicht ausgeführt werden kann, antwortet malloc mit nil/null/0, um zu zeigen, dass es nicht möglich war die Anfrage auszuführen. Falls dies passiert, wird es Blender nicht möglich sein, die Angeforderte Operatione des Anwenders auszuführen. Das wird höchstwahrscheinlich dazu führen, dass Blender sehr langsam arbeitet oder sich beendet. Um dies vorzubeugen kann das System um mehr Speicher erweitert, die Modelle um Details reduziert oder andere Programme geschlossen werden, welche Speicher für Blender freigeben können.

Fenstertypen



Das Fenstertyp Auswahlmenü.

Die Blender Benutzeroberfläche, welche durch das Betriebssystem rechteckige Fenster darstellt, ist in verschiedene rechteckig Fensterrahmen (Window Frames) unterteilt. Jeder Fensterrahmen kann verschiedene Informationen anzeigen, je nachdem was für ein Fenstertyp (Window Type) ausgewählt ist. (In diesem Abschnitt werden die Begriffe Fenster, Frame oder Window als Synonym für einen solchen Rahmen verwendet.)

Jeder Frame ist unabhängig von den anderen und man kann den gleichen Fenstertyp auch in anderen Frames verwenden. So kann man zum Beispiel mehrere 3D-Fenster haben, von denen jedes die aktuelle Szene aus einem anderen Blickwinkel zeigt. Man kann Fenster unterteilen und wieder vereinigen um das Interface an die jeweiligen Aufgaben anzupassen. Außerdem kann man bei einzelnen Fenstern den Header deaktivieren, um den Platz besser zu nutzen.

[Lesen Sie mehr über Organisieren von Frames \(Rahmen\) »](#)

Fenstertypen werden durch ihre Funktion unterschieden:

- [Der 3D View \(3D Ansicht\)](#) - graphische Ansicht der aktuellen Szene.
- [Der Node Editor](#) - zum Bearbeiten von Bildern, Texturen und Materialien sowie Compositing.
- [Der Image/UV Editor](#) - zum Bearbeiten von UV-Maps und Bildern.
- [Der Properties Editor \(Eigenschaftenfenster\)](#) - zeigt einige Eigenschaften des eines ausgewählten Objektes.
- [Der File Browser \(Dateibrowser\)](#) - wird verwendet zum Organisieren, Speichern und Laden von Dateien.
- [Der Outliner \(Objektmanager\)](#) - Hilft beim Suchen und Organisieren von Objekten.
- [Die Timeline \(Zeitleiste\)](#) - Schaltflächen zur Wiedergabesteuerung.
- [Der Graph Editor](#) - zum Verwalten von Animations-Keyframes und deren Inter-/Extrapolation.
- [Der NLA Editor](#) - zum Verwalten von nichtlinearen Animations-Sequenzen.
- [Das Dope Sheet](#) - zum Zusammensetzen von einzelnen Aktionen zu Aktions-Sequenzen.
- [Der Video Sequence Editor \(Video Sequenzeditor\)](#) - zum Zusammensetzen von Videosequenzen.
- [Der Logic Editor \(Logikeditor\)](#) - ein Fenster mit dem die Spielelogik bearbeitet werden kann.
- [Der Text Editor](#) - kann benutzt werden, um Notizen und Dokumentationen zum aktuellen Projekt zu erstellen, oder um Python-Scripts zu schreiben.
- [Die Console \(Konsole\)](#) - zur direkten Benutzung von Python in Blender.
- [User Preferences \(Einstellungen\)](#) - enthält Optionen zum Anpassen von Blender an die persönliche Arbeitsweise und den Computer.

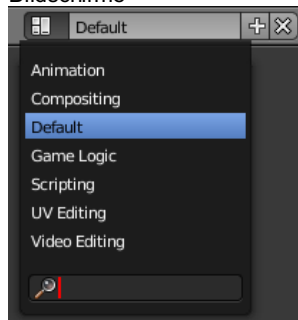
Man kann den Fenstertyp verändern, indem man in der Kopfleiste (Header) auf den ersten Knopf von links klickt. Ein Popup-Menü erscheint, in dem man die verschiedenen Fenstertypen auswählen kann, siehe (*Das Fenstertyp Auswahlmenü*).

Um eine detailliertere Beschreibung der einzelnen Fenstertypen zu erhalten, klicken Sie die obigen Links oder lesen im Referenz-Handbuch, Sektion [III Windows \(eng\)](#) nach.

Sie auch

- [Knöpfe](#)
- [Fenster System](#)

Bildschirme



Layout Liste

Blenders Flexibilität im Umgang mit Fenstern erlaubt es, verschiedenen Arbeitsumgebungen für verschiedene Aufgaben wie Modeling, Animation oder Scripting zu erstellen. Dabei ist es oft nützlich, zwischen den Verschiedenen Umgebungen hin- und herwechseln zu können.

Für jeden dieser wichtigen, kreativen Schritte, hat Blender vordefinierte "Bildschirme", die Ihnen Fenstertypen bereitstellen, die die Arbeit schnell und effizient erleichtern. *Bildschirme* sind wichtige vordefinierte Fensterlayouts. Wenn Sie Probleme haben, einen speziellen Bildschirm zu finden, können sie die Suchfunktion am Boden benutzen (Bild rechts).

Verfügbare Standardbildschirme

Animation

Bewegt Charaktere und andere Objekte, ändert Formen oder Farben etc.

Compositing

Kombiniert verschiedene Teile einer Szene (z.B. Hintergrund, Figuren, Spezialeffekte) und filtert diese (z.B. Farbkorrektur).

Default

Das Standardlayout, dass von neuen Blenderdateien benutzt wird. Ideal um neue Objekte zu modellieren.

Game Logic

Planung und Programmierung von Spielen in Blender.

Scripting

Dokumentation Ihrer Arbeit und/oder schreiben von Benutzerskripten um Blender zu automatisieren.

UV Editing

Stellt eine flache Projektion eines Objekt Meshes in 2D dar, um die Textur auf der Oberfläche kontrollieren.

Video Editing

Schneiden und Bearbeiten von Animationssequenzen.

Blender sortiert diese Bildschirme automatisch alphabetisch oder in numerischer Reihenfolge. Die Liste ist verfügbar in der Info Fenster Kopfleiste, oben im Bildschirm. Die verwirrt gerade Anfänger in Blender, weil es keine Menüleiste ist sondern ein einfaches Fenster in der Kopfleiste.

Um zum nächsten Layout zu schalten, kann man den Shortcut Ctrl→ verwenden; um zum Vorherigen zu wechseln, drücken Sie Ctrl←.



Bildschirm und Szenen Auswahl

Standardmäßig wird für jedes Layout die zuletzt aktive [Szene](#) gespeichert. Das Umschalten zu einem anderen Layout wechselt entsprechen auch zu dieser Szene.

Alle Änderungen der Fenster wie unter [Fenstersystem](#) und [Fenstertypen](#) beschrieben, werden im aktuellen Bildschirm gespeichert. Ändert man ein Fenster in einem Layout, sind andere Layouts nicht betroffen.

Konfiguration der Layouts

Einen neuen Screen hinzufügen

Eine der Optionen im Screen-Dropdownmenü ist **Add New**. Klickt man also (⊕) und wählt ADD NEW aus, wird ein neues Layout auf Basis des aktuellen hinzugefügt.


Am besten vergibt man dafür einen Namen, der mit einer Nummer beginnt, so dass man gezielt mit den Cursortasten hin- und herschalten kann. Man kann ein Layout umbenennen, indem man mit der linken Maustaste LMB (☞) einfach in das Screen-Auswahlfeld klickt und einen neuen Namen eingibt. So könnte man zum Beispiel ein neues Layout "6-MyScreen" anlegen (siehe ([Screen- und Scene-Auswählménüs](#))).

Löschen eines Layouts

Man kann einen Screen löschen, indem man den Delete-datablock-Button (✕) klickt und den Popup-Dialog Delete current screen bestätigt. Siehe ([Screen- und Scene-Auswählménüs](#)).

Fenster neu anordnen

Nun kann man die Fenster verschieben, aufteilen und zusammenfügen bis man das gewünschte Layout erstellt hat. Mit CtrlU wird

dieses Layout in den Standardeinstellungen gespeichert. Das Buttons-Fenster hat eine spezielle Option im Rechtsklickmenü RMB  auf dessen Hintergrund, mit der man festlegt, ob die Panels horizontal (von Links nach Rechts) oder vertikal (von Oben nach Unten) angeordnet werden sollen.

Layouts aus Dateien laden

Wenn man eine .blend-Datei speichert, beinhaltet sie auch die definierten Layouts. Beim Laden einer Datei kann man mit dem Load UI-Button im Header des Dateibrowsers festlegen, ob Blender die Screens aus der Datei laden oder das aktuelle Layout beibehalten soll. Ist Load UI aktiv, werden die Screens der Datei geladen.

Zusätzliche Layouts

Da die Funktionen von Blender immer umfangreicher werden, und man sich mit der Zeit immer besser mit Blender auskennt (je nachdem wofür man Blender genau einsetzt), sollte man darüber nachdenken zusätzliche Layouts anzulegen (um den Workflow möglichst gut abzubilden). Zum Beispiel:

1-Model

4 3D-Fenster, Buttons-Fenster mit Editing-Buttons

2-Lighting

3D-Fenster zum verschieben Lichtquellen, UV/Image-Editor zum Anzeigen gerenderter Bilder, Buttons-Fenster mit Rendering- und Lampen-Einstellungen.

3-Material

Buttons-Fenster mit Materialien-Einstellungen, 3D-Fenster zum auswählen der Objekte, Outliner, Library-Skript (sofern genutzt).

4-UV Layout

UV/Image-Editor, 3D-Fenster zum Bearbeiten von *Seams* und Auswählen von Oberflächen fürs Unwrapp.

5-Painting

UV/Image-Editor für Texture-Painting, 3D-Fenster um direkt auf einem Object zu malen (im UV-Face-Select-Modus), 3 kleine 3D-Fenster mit Referenz-Bildern als Hintergrund, Buttons-Fenster

6-Animation

Ipo-Editor, 3D-Fenster fürs Armature-Posing, NLA-Editor

7-Node

Großes Node-Editor-Fenster, UV/Image-Editor zur Anzeige des gerenderten Bildes.

8-Sequence

Ipo-Editor, VSE-Fenster in Image-Preview-Modus, VSE in Timeline-Modus, ein Timeline-Fenster, und das gute, alte Buttons-Fenster.

9-Notes/Scripting

Outliner, Text-Editor

Layouts wiederverwenden

Wenn man ein Layout definiert hat und es in zukünftigen Datei auch verwenden will, sollte man es einfach in den mit CtrlU in den Standardeinstellungen speichern.

Szenerien

Szenerien sind ein sehr nützliches Werkzeug um Projekte zu managen. Das Würfel Model in einem leeren Raum, welches man beim ersten Start von Blender sieht, ist die standard Szenerie. Szenerien kann man sich vorstellen wie Tabs in einem Webbrowser. Beispielsweise kann der Webbrowser viele Tabs zur gleichen Zeit geöffnet haben. die Tabs können leer sein, identische oder unterschiedliche Sichten der selben Webseite oder gänzlich unterschiedliche Seiten zusammen zeigen. Blenders Szenerien arbeiten in ganz ähnlicher Art und Weise. Dort kann es eine leere Szenerie, eine komplett unabhängige Kopie der Szenerie oder eine neue Kopie des Originals geben, welche wie auch immer daran anknüpft.

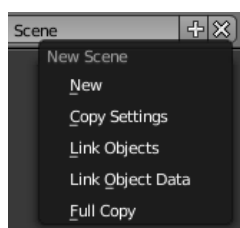
Szenerien können mithilfe des Scene selectors im Info Fensterkopf (Die Leiste am oberen Ende der meisten Blender Layouts. Siehe dazu *Bild- und Szenerienwähler*) ausgewählt werden.




Screen and Scene selectors

Konfiguration der Szenerie

Hinzufügen einer neuen Szene



Add Scene menu

Eine neue Szene kann per Klick auf  in der Szenenwahloption hinzugefügt werden. Dabei kann aus 5 Optionen zu deren Inhalt gewählt werden (*Add Scene menu*).

Um korrekt aus den Optionen wählen zu können ist das Verständnis über "Objekte" und "Objektdaten" von entscheidender Bedeutung. Jedes graphische Element von Blender (Mesh, Lamp, Curve, etc.) ist aus zwei Teilen zusammengestellt: ein Objekt und Objektdaten (auch bekannt als ObData). Das Objekt hält Informationen über die Position, Rotation und Größe von einzelnen Elementen bereit. Die ObData hält Informationen über Maschen, Materiallisten usw. fest. ObData ist gewöhnlich an jede Instanz eines speziellen Elementtyps geknüpft. Jedes Objekt hat einen Link zu dessen assoziierten ObData. Eine einzelne ObData kann auch von vielen Objekten verwendet werden.

Die fünf Optionen, welche abhängig machen, wie viel Informationen aus der aktuellen Szenerie *kopiert* und wie viele *gemeinschaftlich verwendet* ("verlinkt") werden:

New

Kreiert eine leere Szenerie. In der neuen Szene sind die Renderoptionen auf Standardwerte gesetzt.


Copy Settings

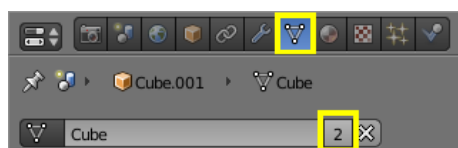
Kreiert eine leere Szene analog der "New" Auswahl, kopiert jedoch auch die Renderoptionen von der Originalen Szene in die Neue.

Link Objects

Sie ist die oberflächliste aller wählbaren Kopiermöglichkeiten. Diese Option kreiert die neue Szene mit den selben Inhalten wie der derzeit gewählten Szene. Anstelle von Kopieren der Objekte enthält die neue Szene nur Links zu den Objekten der alten Szene auf der Objektebene. So werden Änderungen in der neuen Szene auch dieselben Änderungen im Original herbeiführen, da die Objekte buchstäblich die selben sind. Das selbe funktioniert auch umgekehrt bei Änderungen in der Originalszene.

Link Object Data

Kreiert neue, duplizierte Kopien aller Objekte der derzeit gewählten Szene, jedoch hat jedes der duplizierten Objekte Links zu den ObData der korrespondierenden Objekte in der Originalszene. Das bedeutet, dass die Position, Orientierung und Größe der Objekte in der neuen Szene geändert werden können, ohne andere Szenen zu beeinflussen. Jedoch beeinflusst jede andere Modifikation der ObData andere Szenen. Dies rührt daher, das jetzt eine einzige Instanz der ObData von allen korrespondierenden Objekten der Szenen gemeinschaftlich verwendet wird. Soll eine Änderung unabhängig anderer Szenen vorgenommen werden, muss das Objekt manuell in der neuen Szene erstellt werden. Eine "Einzelnutzerkopie" kann per LMB  auf die Nummer in Object Data Panel des [Eigenschaften Fensters](#), erstellt werden. Das hat den Effekt einer neuen, unabhängigen Kopie der ObData.



Full Copy

Ist die tiefgründigste wählbare Kopierform. Nichts wird dabei gemeinschaftlich verwendet. Diese Option kreiert eine völlig unabhängige Szenerie mit Kopien des Inhaltes der derzeitig selektierten Szene. Jedes Objekt in der Originalszene wird dupliziert und ein Duplikat der ObData wird ebenso angefertigt.

Um Blenders Umgang mit Daten besser zu verstehen wird an dieser Stelle auf [Blender's Library and Data System](#). verwiesen.

Ein kurzes Beispiel

Man stelle sich eine Bar-Szene in einem Film vor. Die Bar wird initial als eine saubere Version erstellt, bei dem alles unverletzt und geordnet steht. Man entscheidet sich für das Erstellen einer Aktion in einer separaten Szenerie. Die Aktion in der Szene wird zeigen, welcher Typ des linkens am zutreffendsten ist:

Link Objects

Jedes Objekt wird zu der Originalszene verlinkt. Wird die Platzierung einer Wand korrigiert, geschieht diese Änderung in jeder Szene mit der Bar als Einstellung.

Link Object Data


Wird nützlich sein, wenn die Positionen von Objekten verändert werden sollen, jedoch deren Form- und Materialeinstellungen konstant bleiben sollen. Beispielsweise stehen Stühle möglicherweise auf dem Boden der "besuchte Bar"-Szene und auf den Tischen in der "wir schließen"-Szene. Verändern die Stühle nicht ihre Form, ist es nicht nötig Speicher für exakte Kopien zu verschwenden.

Full Copy

Ein Glas, welches am Boden zerbricht benötigt eine eigene Kopie, da die Masche die Form ändert.

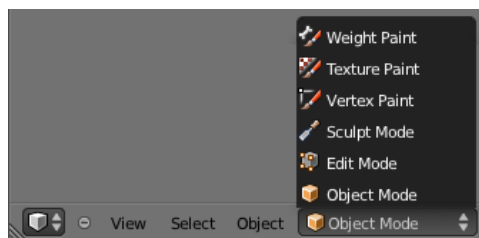
Es ist nicht möglich alles oben beschriebene in der selben Szene anzuwenden, jedoch könnte es beim Verstehen helfen, warum verschiedene Objekte auf verschiedene Arten verknüpft werden.

Eine Szene löschen

Eine Szene kann per Klick auf den Delete datablock-Knopf im Szenenwähler gelöscht werden (). Siehe dazu *Bild- und Szenenwähler*.

Modi

Modi sind ein objektorientiertes Feature von Blender. Dies bedeutet, dass sich die gesamte Blender Anwendung immer in einem einzigen Modus befindet und somit die verfügbaren Modi abhängig des aktuell gewählten, aktiven Objekttyp variieren. Die meisten aktivieren lediglich den standardmäßigen Objektmodus, wie Kameras, Lampen usw. Jeder Modus ist designed worden, um einen Aspekt des selektierten Objekts zu editieren. Details sind in der folgenden Tabelle (*Blenders Modi*) zu sehen.



Beispiel für die Wahl eines Modus (Maschenobjekt).

Der aktuelle Modus wird in der Mode drop-down Liste des 3D View Headers gesetzt (siehe dazu *Beispiel für die Wahl eines Modus (Maschenobjekt)*).



Objekte können nur im Object-Modus selektiert werden. In allen anderen Modi wird die aktuelle Selektion gesperrt (eine Ausnahme ergibt das Arbeiten im Pose-Modus mit Armaturen.)

Modi können einige Dinge in Blender beeinflussen:

- Sie können die wählbaren Panels und/oder Einstellungen einiger Schaltflächen Kontexte beeinflussen.
- Sie können das Verhalten ganzer Fenster, wie beispielsweise den UV/Image Editor (und offensichtlich auch 3D Views), beeinflussen.
- Sie können die verfügbaren Header-Werkzeuge verändern, sowohl Menüs und/oder Menüeinträge, als auch andere Einstellungen. Beispielsweise ändert sich das Object menu im Object mode des 3D Views in ein Mesh menu im Edit mode (inklusive einem aktiven Maschenobjekt!), in ein Paint menu im Vertex Paint mode, ...

Blenders Modi

Icon	Name	Kürzel	Beschreibung
	Object mode	<i>keiner</i> ¹	Der Standardmodus, der für alle Objekttypen wählbar ist. Er ist für die Editierung des Objekt-Datenblockes (z.B. Position/Rotation/Größe) bestimmt.
	Edit mode	Tab ¹	Ein Modus der für alle renderbaren Objekttypen verfügbar ist. Er ist für deren "Form", Editierung des ObData Datenblockes (z.B. Kanten/Ecken/Flächen für Maschen, Kontrollpunkte für Kurven/Oberflächen, usw.) zuständig.
	Sculpt mode	<i>keiner</i> ¹	Ein nur-Maschen-Modus, welcher Blenders 3D-sculpting Werkzeug aktiviert.
	Vertex Paint mode	<i>keiner</i> ¹	Ein nur-Maschen-Modus, welcher es erlaubt die Eckpunkte einzufärben.
	Texture Paint mode	<i>keiner</i> ¹	Ein nur-Maschen-Modus, welcher es erlaubt die Textur direkt am Modell in der 3D-Ansicht einzufärben.
	Weight Paint mode	Ctrl+Tab ²	Ein nur-Maschen-Modus, der für die Wichtung von Eckpunktgruppen zuständig ist.
	Particle mode	<i>keiner</i> ¹	Ein nur-Maschen-Modus, der für Partikelsysteme zuständig ist. Dieser ist besonders nützlich mit editierbaren Systemen, wie Haar.
	Pose mode	Ctrl+Tab ²	Ein nur-Armaturen-Modus, der für das Positionieren von Armaturen zuständig ist.

Anmerkungen zu Modi Kürzeln:

1. Tab ruft den Edit mode auf.
2. Ctrl+Tab wechselt zwischen den Modi Weight Paint (meshes)/Pose (armatures) und dem aktuell gewählten. Das Kürzel kann andere, interne Bedeutungen in einigen Modi haben. Im Sculpt mode wird es beispielsweise dafür verwendet den aktuellen Pinsel zu wählen.

Die Verwendung von Kürzeln kann also, vor Allem beim Arbeiten mit Maschen, ziemlich trickreich sein.

An dieser Stelle wird die Beschreibung weiterer Modi vernachlässigt. Die Meisten werden aufgrund der Relevanz für das Thema im [Modellierungskapitel](#) behandelt. Der Particle mode wird in der [Partikelsektion](#) näher erläutert, während die Modi Pose und Edit für Armaturen beim [Rigging](#) erklärt werden.

Anmerkung

Falls in diesem Handbuch eine Schaltfläche oder Menüoption beschrieben wird, die im eigenen Programm nicht erscheinen, so kann es sein, dass der falsche Modus gewählt und die Option dafür nicht valide ist.










Kontexte - Zusammenhänge

Der Eigenschafts (oder Knopf) Fenster zeigt einige Kontexte, die über die Symbolleiste im Kopf gewählt werden können (siehe *Kontextknöpfe Beispiel*).



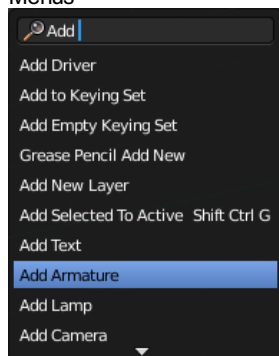
Kontextknöpfe Beispiel

Die Anzahl und der Typ von Knöpfen ändert sich mit dem ausgewählten Zusammenhang, so dass nur nützliche Knöpfe auftauchen. Die Ordnung dieser Knöpfe folgt einer Hierarchie, über die unten ausführlich berichtet wird:


-  [Rendern](#): Alles was die Rendereingabe betrifft (Dimensionen, Antialiasing, Leistung usw.).
-  [Szene](#): Gravitation der Szene, den Einheiten und anderen allgemeinen Information.
 - [Welt](#): Umgebungsbeleuchtung, Himmel, Nebel, Sterne und Ambient Occlusion.
 - [Objekt](#): Transformationen, Anzeigeoptionen, Sichtbarkeitseinstellungen (über Ebenen) Verdoppelungseinstellungen und Animations-Information (bezüglich der Objektposition).
 -  [Objektbeschränkungen](#): Wird verwendet um ein Objekttransformationen (Position, Rotation, Skalierung), Tracking und Beziehungseigenschaften zu kontrollieren.
 -  [Objektmodifikatoren](#): Operations that can non-destructively affect Objects by changing how they are rendered and displayed without altering their geometry (e.g. mirror and smoothing).
 -  [Objektdaten](#): Enthält die ganzen Objekt spezifische Daten (Farbe einer Lampe, Brennweite der Kamera, Vertex-Gruppen usw.). Das Icon unterscheidet sich mit dem Typ des Objektes (ein hier gezeigter ist das Mesh-Objekt).
 -  [Material](#): Informationen über die Oberfläche (Farbe, Spiegelung, Transparenz, etc).
 -  [Texturen](#): Verwendet durch Materialien, um zusätzliche Details (z.B Farbe, Durchsichtigkeit, unechte 3-dimensionale Tiefe) zur Verfügung zu stellen.
 -  [Partikel](#): Verwendet durch Materialien, um zusätzliche Details (z.B Farbe, Durchsichtigkeit, unechte 3-dimensionale Tiefe) zur Verfügung zu stellen.
 -  [Physik](#): Eigenschaften in Zusammenhang mit Stoff, Kraftfelder, Kollision, Flüssigkeit und Rauch-Simulation.


Die [Knöpfe](#) in jedem Kontext sind gruppiert in [Paneele](#).

Menüs



Das Space-Menü

Blender beinhaltet einige Menüs, jedes diese ist über eine Fenstermenüleiste oder direkt durch [HotKeys](#) an der Mausposition oder durch klicken der RMB  auf einem Fensterrand, einem Knopf oder woanders auf dem Bildschirm, zu erreichen. Ein sensitives Kontextmenü wird angezeigt, wenn es etwas verfügbares für dieses Interface-Element gibt.

Zusätzlich ist ein Menü mit dem Zugriff auf alle Blender Befehle durch drücken der Space Taste verfügbar (siehe *Das Space-Menü*). Geben Sie einfach den Namen des Befehls ein und die Suchfunktion zeigt Ihnen das Ergebnis an. Wenn die Liste eine engere Auswahl anzeigt, können Sie mit der LMB  auf den gewünschten Befehl klicken oder durch ↓ und ↑ auswählen und mit Return bestätigen.

Wenn Sie das alte Toolbox-Menü aus der Version 2.4x vermissen, können Sie ein ähnliches über das 3D View: Dynamic Spacebar Menu Add-On hinzufügen. Dieses kann mit dem Add-Ons Reiter, des Preferences Window (Benutzer Einstellungen Fenster), installiert werden.

[Lesen Sie mehr über das Installieren von Add-Ons »](#)

Einige Menüs haben besondere Zusammenhänge und sind nur unter bestimmten Situationen verfügbar. Zum Beispiel das Specials Menü (W Taste) ist nur im 3D Fenster, bei aktivem Edit Mode (Bearbeitungsmodus), verfügbar.

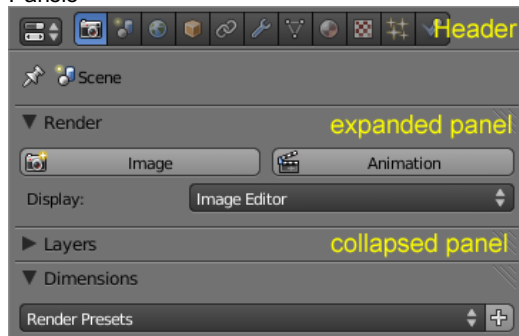
Während Sie mit Blender arbeiten, sollten Sie wissen welcher Modus aktiv ist und welches Objekt ausgewählt ist. Dies hilft zu wissen, welche Hotkeys wann und wie arbeiten.



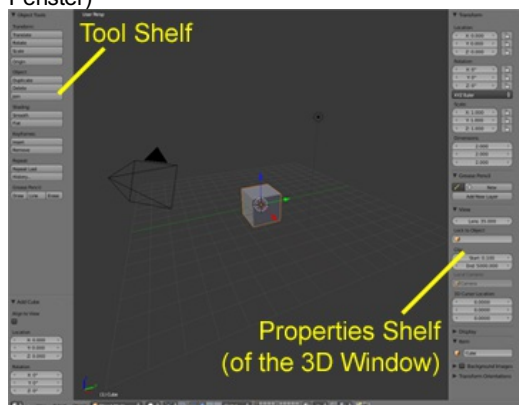
Menüs auf dem Mac

Weil Blender nicht das standard Menüsystem von Betriebssystemen verwendet, haben Sie bei einem Mac oben eine überflüssige Menüleiste. Um diese zu entfernen, siehe [diesen Post](#) auf Macworld, aber seien Sie vorsichtig, dieser ist sehr Komplex. Als Alternative: einfach Blender mit dem letzten Knopf im Info-Fenster (meistens oben im Fensterlayout), in den Vollbild Modus wechseln.

Panele



Teile des Properties Window (Eigenschaften Fenster)



Shelves (Ablage) im 3D Fenster

Panele befinden sich im Allgemein im Properties Window (Köpfe-Fenster in der Version 2.4x), diese sind auf der rechten Seite des standard Fensterlayouts zu finden (siehe *Teile des Properties Window (Eigenschaften Fenster)*).

Panele sind im Tool shelf (Werkzeug Ablage) und dem Properties Shelf (Eigenschaften Ablage) zu finden, diese sind einschaltbare Teile des 3D Fensters. Um das Tool Shelf anzuzeigen, benutzen Sie View » Tool oder drücken Sie die Taste T. Um das Properties Shelf anzuzeigen, benutzen Sie View » Properties oder drücken die Taste N. Siehe *Shelves im 3D Fenster*.

Das Properties Window (Eigenschaften Fenster) beinhaltet eine Kopfleiste in der einige [Kontexte](#) gewählt werden können. Jeder Kontext hat eine unterschiedliche Anzahl an Paneltypen. Zum Beispiel das Render Kontext besitzt Panele die Ihnen erlauben die Dimensionen und das Anti-Aliasing der Renderausgabe zu ändern, während der Material Kontext Panele besitzt, die Ihnen erlauben eine Farbe festzulegen, Transparenzen, Texturen etc.

Panele im Properties Window (Eigenschaften Fenster) können vertikal oder horizontal mit dem Klick der RMB auf dem Properties Window und Wählen der Ausrichtung im Menü, ausgerichtet werden. Bedenken Sie, dass die Panele im Properties Window für eine vertikale Ausrichtung optimiert sind. Eine horizontale Ausrichtung könnte die Arbeit damit erschweren.

Die Anordnung und die Ansicht des Panels können in den Einstellungen geändert werden. Zum Beispiel können Panele:

- im Fenster (oder in der Ablage (shelve)) mit dem Klicken, halten und ziehen der LMB auf der rechten oberen Ecke (ähnlich dem Rahmenteiler, es hat 3 Linien in einem Dreieck) des Widget, verschoben werden.
- mit dem Wheel hoch und runter gescrollt werden.
- durch halten von Ctrl MMB und bewegen der Maus nach rechts und links, herein bzw. heraus gezoomt werden.
- mit dem Klick der LMB auf das schwarze Dreieck auf der linken Seite einer Überschrift, eingeklappt oder ausgeklappt werden.

Weitere Details über jedes Panel finden Sie im [Panel](#) Abschnitt, oder in bestimmten Abschnitten des Handbuchs.


Knöpfe und Steuerelemente

Knöpfe und andere Steuerlemente können überall in jedem [Fenster](#) der Blender Benutzeroberfläche gefunden werden. Die verschiedenen Typen von Steuerungen werden unten beschrieben.

Operationsknöpfe

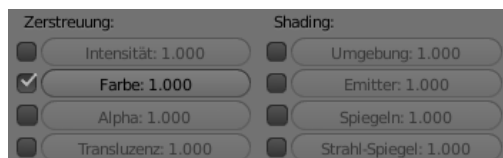


Operationsknöpfe

Das sind Knöpfe, die eine Operation ausführen, wenn mit LMB  darauf geklickt wird. Sie können sie durch ihre graue Farbe im Standard Blender-Schema erkennen.

Das Drücken von StrgC über diese Knöpfe kopiert ihren Python-Befehl in die Zwischenablage, die in der Python-Konsole oder im Texteditor verwendet werden kann, wenn man Skripte schreibt.

Schaltbare Knöpfe



Schaltbare Knöpfe

Schaltbare Knöpfe bestehen aus Checkboxes. Das Klicken auf diesen Typ des Knopfes schaltet den Zustand um, aber führt keine Operation aus. In einigen Fällen wird der Knopf einem Nummernknopf beigelegt, um den Einfluss der Eigenschaften zu kontrollieren.

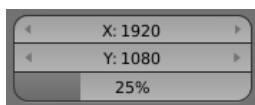
Auswahlknöpfe



Auswahlknöpfe







Auswahlknöpfe werden verwendet, um eine Option aus Vielen auszuwählen.



Nummernknöpfe



Nummernknöpfe

Nummernknöpfe können durch ihre Beschriftung identifiziert werden, der in den meisten Fällen den Namen und einen von einer Zahl gefolgt Doppelpunkt enthält. Nummernknöpfe werden auf mehrere Weisen behandelt:

1. Um einen Wert in Schritten zu ändern, klicken Sie mit LMB  auf die kleinen Dreiecke an der Seite des Knopfes.
2. Um einen Wert in größeren Abständen zu ändern, halten Sie die LMB  gedrückt und ziehen die Maus nach links oder rechts. Wenn Sie Strg nach dem Drücken der LMB  halten, wird der Wert in getrennten Schritten geändert; wenn Sie  Shift statt dessen halten, erhalten Sie eine feinere Kontrolle über die Werte.
3.  Enter oder LMB  ermöglicht die Eingabe per Hand.

Wenn Sie Werte per Hand eingeben, drücken Sie  Home oder  End, um den Cursor an den Anfang oder an das Ende der Eingabe zu bewegen. Durch Drücken von Esc brechen Sie die Eingabe ab. Sie können Werte eines Knopfes, durch überfahren der Maus und drücken von StrgC, kopieren. Auf die gleiche Weise fügen Sie Werte mit StrgV ein.

Ausdrücke

Sie können auch Ausdrücke wie $3*2$ anstelle von 6 eingeben, oder $5/10+3$. Jede Konstante wie π (3.142) oder Funktionen wie $\text{sqrt}(2)$ (Quadratwuzel aus 2) können verwendet werden.

Diese Ausdrücke werden von Python ausgewertet, alle verfügbaren mathematischen Ausdrücken finden Sie hier: [Mathemodul Referenz](#)

Einheiten

Sowie Ausdrücke können Sie Einheiten mit Zahlen mischen. Um damit zu arbeiten, müssen sie in den Szene-Einstellungen (Metrisch oder Imperial) gesetzt werden.

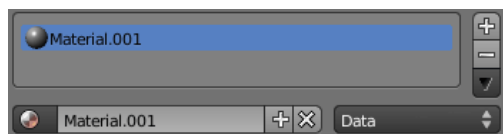
Gültige Einheiten schließen ein...

- 1cm
- 1m 3mm
- 1m, 3mm
- 2ft
- 3ft/0.5km

- 2.2mm + 5' / 3" - 2yards

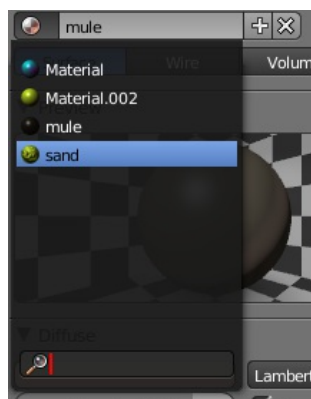
Bemerken Sie, dass die Kommata optional sind, bemerken Sie auch, wie Sie sie zwischen metrischem und imperialen vermischen können, auch wenn die Anzeige nur einer nach dem anderen zeigen kann.

Menüknöpfe



Datenblock-Verbindungsknopf

Verwenden Sie die Menüknöpfe, um mit Dingen in dynamisch erstellten Listen zu arbeiten. Menüknöpfe werden hauptsächlich verwendet, um Datenblöcke mit einander zu verbinden. Datenblöcke sind Dinge wie Meshes, Objekte, Materialien, Texturen und so weiter. Die Verbindung eines Materials mit einem Objekt bestimmt dieses Material für die ausgewählten Objekte.



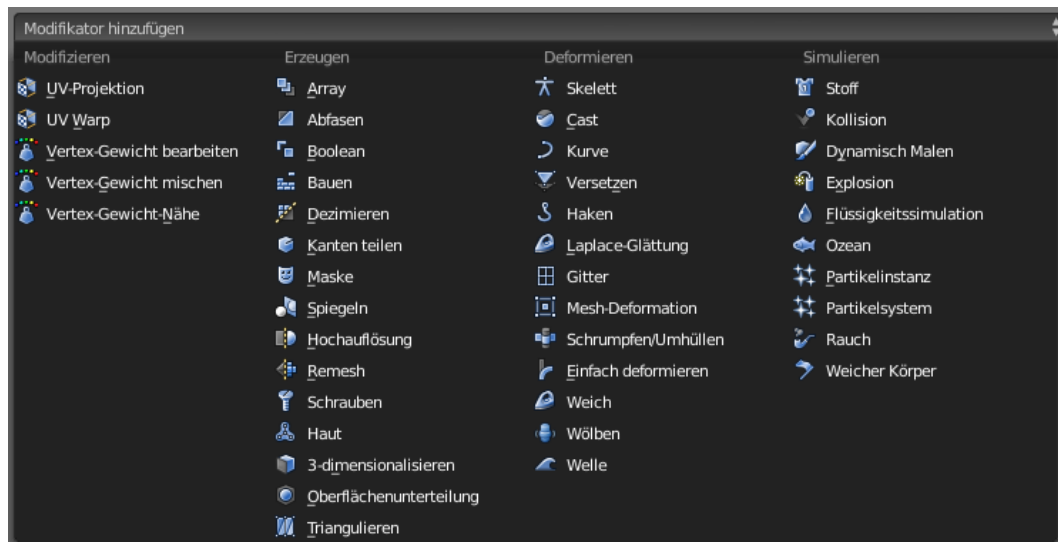
Datenblock-Verbindungs Menü mit Suche

1. Der erste Knopf (mit dem Icon des Typs DataBlock) öffnet ein Menü, in dem Sie Datenblöcke auswählen können, um diese durch Klicken der LMB auf dem gewünschten Gegenstand zu verbinden. Diese Liste hat ein Suchfeld am Boden.
2. Der zweite Knopf zeigt den Namen des verbundenen Datenblocks an und lässt Sie ihn, nach dem Klicken der LMB, bearbeiten.
3. Der "+" Knopf dupliziert einen gegenwärtigen Datenblock und wendet ihn an.
4. Der "X" Knopf löscht die Verbindung.

Manchmal gibt es eine Liste von angewandtem Datenblöcken (wie eine Liste der auf dem Gegenstand verwendeten Materialien). Siehe *Datenblock-Verbindungsknopf* oben.

1. Um einen Datenblock auszuwählen, klicken Sie mit der LMB auf ihn.
2. Um einen neuen Abschnitt hinzuzufügen (z.B. Material, Partikel Systeme etc.) klicken Sie LMB auf den "+" Knopf rechts in der Liste.
3. Um einen Abschnitt zu entfernen, klicken Sie LMB auf das "-" rechts in der Liste.

Ein anderer Typ eines Menüknopf Blockes zeigt eine statische Liste mit einer Reihe von Optionen. Zum Beispiel öffnet Modifikator hinzufügen ein Menü mit allen verfügbaren Modifikatoren.



Modifikator Optionen

Nicht verbundene Objekte

Nicht verknüpfte Daten gehen **nicht verloren, bis Sie Blender verlassen**. Das ist eine mächtige Rückgängig Funktion. Wenn Sie ein Objekt löschen, wird die Verbindung, des ihm zugeteilte Material gelöst, es aber ist immer noch da! Sie müssen es wieder mit einem anderen Objekt oder es mit einem "Unechten Benutzer" verbinden (d. h. indem Sie auf diese Auswahl im entsprechendem Datenblock in der Datablock-Ansicht vom Outliner klicken)

[Lesen Sie mehr über unechte Benutzer »](#)

Farbwähler Steuerelement

Einige Steuerelemente öffnen ein Dialogpanel. Zum Beispiel, Farb Steuerelemente öffnen den Farbwähler Dialog wenn man draufklickt. Siehe *Farbwähler*.




Farbwähler

- ← Backspace setzt die Farbe zur Vorgabe zurück.
- Mouse wheel ändert die Helligkeit.

Pipette

Die Pipette erlaubt Ihnen eine Farbe überall im Blenderfenster aufzunehmen.

LMB  und das Ziehen der Pipette mischt die Farben, über die Sie ziehen. Dies kann bei der Farbaufnahme rauschender Bilder helfen. Spacebar startet wieder das Mischen von Farben und setzt es wieder zurück.

Verschachtelte Knöpfe

Manchmal offenbaren einige Knöpfe zusätzliche Knöpfe. Zum Beispiel hat das Rampen Panel einen Verschachtelten Knopf mit dem Namen Ramp, der zusätzliche Knöpfe offenbart, die sich mit Farbregelung befassen. Siehe *Farbregler vorher* und *Farbregler danach*.



Farbregler vorher



Farbregler danach

Blender Internationalisierung

Seit der Version 2.60 unterstützt Blender internationale Schriftarten und eine Reihe von Sprachoptionen für Benutzeroberfläche und Tooltips. Um diese zu benutzen, öffnen Sie das Fenster User Preferences, die Registerkarte System und schalten die International Fonts Option in der rechten unteren Ecke ein.

Diese zeigt drei Einstellungen:



Internationale Schriftarten im User Preferences Fenster aktivieren.

Language

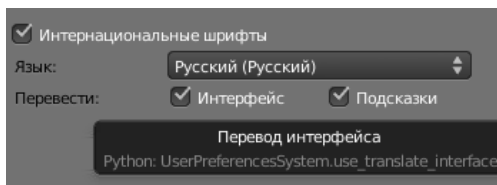
Drop-Down Menü in dem die bevorzugte Sprache gewählt werden kann.

Interface

um die Benutzeroberfläche selbst zu übersetzen (z.B. Steuerlemente und Menüs).

Tooltips

um die Tooltips zu übersetzen.



Blender mit der russischen Sprache aktivieren für Benutzeroberfläche und Tooltips.

Blender aktualisiert den Bildschirm nach dem Auswählen von Interface oder Tooltips (oder beide), um die neue Sprache anzuzeigen. Beachten Sie, dass einige Sprachübersetzungen noch nicht abgeschlossen sind. Der Fortschritt jeder Übersetzung wird im Drop-down Menü angezeigt.



Tip

Da die Mehrheit von Tutorials mit einer englischen Benutzeroberfläche verfasst wurden, kann es hilfreich sein, die Benutzeroberfläche in Englisch zu belassen und nur die Tooltips zu übersetzen.

Schnelles Rendering

Was ist Rendering?

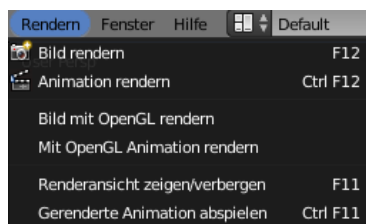
Rendering ist der Prozess des Erstellens eines 2D Bildes. Blender erstellt dieses Bild durch Aufnahme ihres Models und all Ihren Materialien, Texturen, Beleuchtungen und Compositing.

- Es existieren zwei Haupttypen an Render Engines die in Blender eingebaut sind, einer für *Komplettes Rendern*, und ein anderer für *OpenGL Render*. Diese Seite liefert Ihnen grundlegende Informationen zum Rendern von Bildern. Für tieferes Wissen über das *Komplettes Rendern* in Blender, Blender Intern genannt, sehen Sie sich die Seite [Rendering mit Blender Intern](#) an.
- Es existiert ebenfalls ein Abschnitt in diesem Wiki Handbuch, der dem [Cycles](#) Rendering gewidmet ist, welches seit Version 2.61 in Blender enthalten ist.

Rendering eines Bildes mittels *Komplettem Rendern* - Blender Intern

Modus: Alle Modi

Tastenkombination: F12



Kopf des Info Fensters

Um *Komplettes Rendern* mit Blender Intern zu erreichen, können Sie jede der folgenden Optionen verwenden:

- Drücken von F12
- Gehen Sie zu Eigenschaften Fenster » Render Kontext » Render Panel und drücken Sie den Bild Knopf
- Gehen Sie zu Render » Bild rendern vom Kopf aus des Info Fensters (Siehe: *Kopf des Info Fensters*)
- Mittels Blender Suche: Drücken Sie Leertaste, tippen Sie Render und klicken Sie auf Render.

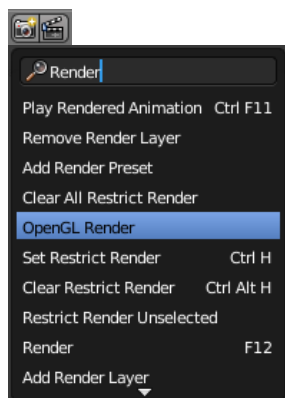
Zum Abbrechen oder Beenden des Render-Vorgangs, drücken Sie Esc.

Rendering eines Bildes mittels *OpenGL Render*

Modus: Alle Modi

Tastenkombination: undefiniert - Sie können eine zugehörige Taste zu Ihrer [Tastenbelegung](#) » hinzufügen

Um einen *OpenGL Render* Vorgang zu starten, können Sie jede der folgenden Optionen verwenden:



Search functionality

- Durch Klicken auf *Aktives Ansichtsfenster mit OpenGL rendern*, in der Kopfzeile der 3D Ansicht, mittels des kleinen Symbols das eine *Kamera* zeigt (zusammen mit einem anderem Symbol das eine *Filmklappe* zeigt).
- Gehen Sie zu Rendern » Bild mit OpenGL rendern von der Kopfzeile des Info Fensters aus (Siehe Bild: *Kopf des Info Fensters*)
- Mittels Blender Suche: Drücken Sie Leertaste, tippen Sie Render und klicken Sie auf OpenGL Render.

Zum Abbrechen oder Beenden des Render-Vorgangs, drücken Sie Esc.

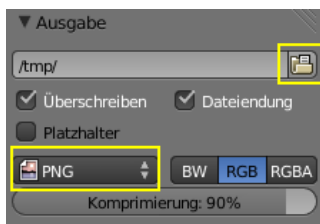
Anpassung der Auflösung



Dimensionen Panel

Das Dimensionen Panel vom Render Kontext erlaubt Ihnen die Auflösung zu ändern. Die Standardeinstellung von Blender ist initial auf **50%** von **1920 x 1080** gesetzt, was ein **960 x 540** Bild ergibt (Hervorgehoben in Gelb, auf dem Bild zum Dimensionen Panel). Höhere Auflösungen und höhere Prozentskalierungen zeigen mehr Details, aber werden mehr Zeit zum Rendern benötigen.

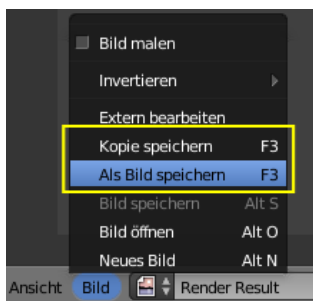
Ausgabeformat und Ausgabedatei



Ausgabe Panel

Sie können auch ein Ausgabeformat und den Ausgabeort für Ihr gerendertes Bild oder Ihre Animation angeben. Standardmäßig werden diese im temporärem Verzeichnis (/tmp), mittels eines absoluten Pfades gespeichert. Sie können Ihre Datei Pfade durch die Anleitung unter [Datei Einrichtungs Kapitel](#) einrichten und Sie können nichtsdestotrotz einen anderen Ordner durch das Klicken auf das Verzeichnissymbol im Ausgabe Panel wählen. Sie können den Typ des Bild- oder Film-Formates für Ihre Arbeit über den Menü-Knopf wählen.

Speichern Ihres Bildes



Save as dialog

Blender speichert Ihr generiertes Bild nicht automatisch. Um Ihr Bild zu speichern, können Sie entweder F3 drücken oder auf Als Bild speichern klicken vom Bild Menü der Kopfzeile des UV/Bild Bearbeiter Fensters. Diese Aktion wird die Blender-interne Dateiverwaltung öffnen, wo sie dann nach Ordnern zum Speichern Ihres generierten Bildes suchen können.

Rendering einer Animation mittels *Komplettem Render* - Blender Intern

Modus: Alle Modi

Tastenkombination: StrgF12



Dimensions panel

Das Rendern einer Animation ist einfach, der Framebereich (hervorgehoben in Rot, im Dimensionen Panel Bild) im Ausgabe Panel wird zur Angabe der **Anzahl an Bildern** die Ihre Animation rendern wird verwendet. Die **Zeit** wird durch die *Bildern-pro-Sekunde* bestimmt, die in der Framerate-Auswahlbox definiert ist (hervorgehoben in Blau, im Dimensionen Panel Bild). Standardmäßig sind **24 FPS** und **250** Frames gesetzt.

Ein kurzes Beispiel zum Verstehen dieser Nummern:

- Das Panel zeigt dass die Animation beim Frame **1** beginnt und beim Frame **250** endet und die Bilder-pro-Sekunde auf **24** gesetzt sind, womit Ihnen die Standardeinstellung von Blender ungefähr **10** (zehn) Sekunden an Animation zeigt {{{1}}}

Um eine Animation mittels *Komplettem Rendern* mit der Blender-internen Engine zu rendern, können Sie jede der folgenden Optionen wählen:

- Drücken Sie StrgF12
- Gehen Sie auf Eigenschaften Fenster » Render Kontext » Render Panel und drücken Sie den Animationsknopf oder
- Gehen Sie auf Rendern » Animation rendern von dem Kopf des Info Fensters aus (Siehe Bild: *Kopf des Info Fensters*)

Zum Abbrechen oder Beenden des Rendering-Vorgangs der Animation, drücken Sie Esc.

Rendern einer Animation mittels *OpenGL Render*

Modus: Alle Modi

Tastenkombination: undefiniert - Sie können eine zugehörige Taste zu Ihrer [Tastenbelegung](#) » hinzufügen

Um eine Animation mittels *OpenGL Render* zu rendern, können Sie eine der folgenden Optionen ausführen:



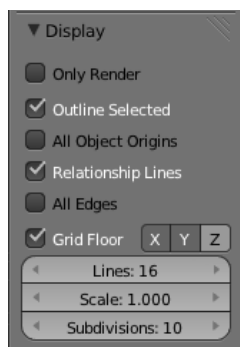
- Durch das Klicken auf das kleine Symbol dass eine *Filmklappe* zeigt (Zusammen mit einem kleinem Bild dass eine *Kamera* zeigt) in der Kopfzeile der 3D Ansicht.
- Durch gehen auf Rendern » Mit OpenGL Animation rendern von der Kopfzeile des Info Fensters aus (siehe Bild: *Kopfzeile des Info Fensters*)

Zum Abbrechen oder Beenden des Rendering-Vorgangs der Animation, drücken Sie Esc.

Nur renderbare Objekte anzeigen

Modus: Alle Modi

Tastenkombination: undefiniert - Sie können eine zugehörige Taste zu Ihrer [Tastenbelegung](#) » hinzufügen



Transform Panel - Display Tab.

Während des Render-Vorgangs (entweder Komplette oder OpenGL), gibt es einige Objekte in der Szene die nicht gerendert werden, entweder wegen ihrem Typ (Knochen, leere Objekte, Kameras, usw.), weil sie ungültig sind oder keine sichtbare Geometrie aufweisen (Mesh ohne einen Vertex, nicht-extrudierte Kurven, usw.), oder weil sie einfach als nicht renderbar festgelegt worden sind.

Blender hat eine Einstellung, um nur Objekte in der Szene anzuzeigen, die auch tatsächlich gerendert werden.

Um auf diese Option zuzugreifen, fahren Sie mit Ihrer Maus zur 3D Ansicht (Fokus darauf), benutzen Sie die Taste N oder klicken Sie

auf das Pluszeichen (+) an der oberen rechten Seite, um das Transformierung-Panel zu zeigen. Beim Durchrollen der Optionen werden Sie den Reiter Anzeige finden, dessen Optionen einstellen, wie Objekte der 3D-Ansicht angezeigt werden.

Aktivieren Sie einfach die Option Nur Rendern - jetzt werden nur Objekte angezeigt die auch wirklich gerendert werden (siehe Bild: Transformierungs-Panel - Anzeige-Reiter). Diese Option funktioniert auch wenn Bilder mittels der OpenGL Renders generiert werden. Beachten Sie das alle anderen Optionen für selektives Anzeigen deaktiviert werden.

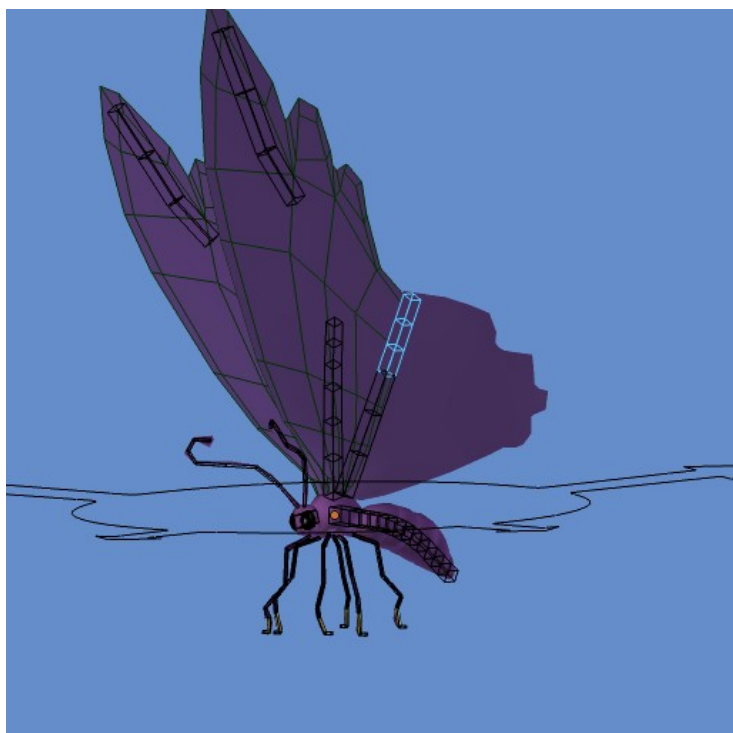
Die Gründe für OpenGL Rendering

OpenGL Rendering erlaubt es dem Animateur, schnell seine Animationen (für Dinge wie Objektbewegungen, alternative Winkel, usw.) zu untersuchen, durch das Hinzufügen eines Entwurfes für qualitatives Rendering des aktuellen Viewports.

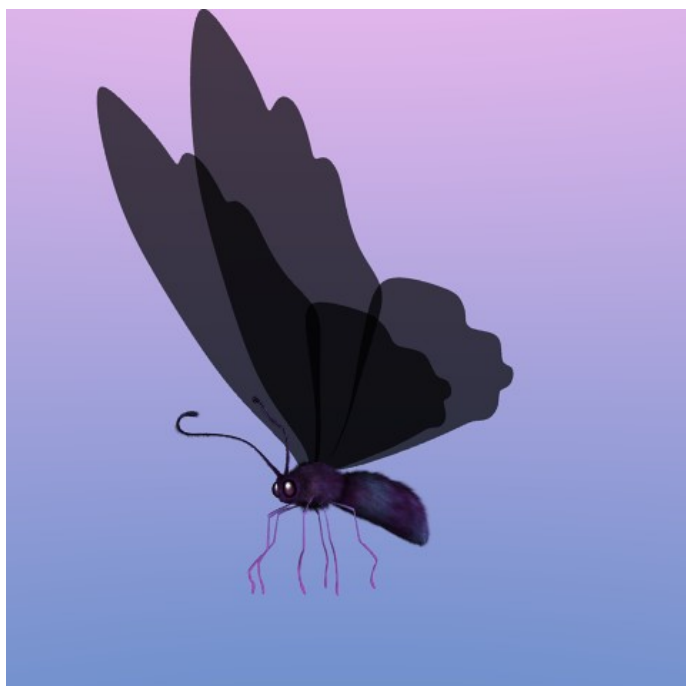
Da er nur mittels OpenGL gerendert wird, kann er sehr viel schneller generiert werden, auch wenn es nur so gut aussieht wie das was sie in der 3D-Ansicht sehen.

Dies erlaubt dem Animateur eine Vorschau seiner Animation mit flüssiger Wiedergabe zu erhalten, die er wegen der Komplexität der Szene mit anderen Mitteln nicht in Echtzeit erhalten könnte (zB: Das Drücken von AltA resultiert in einer zu geringen Anzahl an *Bildern pro Sekunde* um ein gutes Gefühl für die Animation zu erhalten).

Das ist ein Beispiel für ein Bild das mit OpenGL gerendert worden ist:



Und hier ist eins das mit der *kompletten Render* Methode von Blender Intern gerendert worden ist:



Sie können OpenGL um Bilder und Animationen zu Rendern, und Dimensionen mittels den oben angegebenen Schritten verändern.

So wie bei dem normalem Render-Vorgang, können Sie diesen mit Esc abbrechen.

Zurückkehren von Missgeschicken und Problemen

Blender bietet eine Reihe an Wegen an von Missgeschicken vom Anwender zurückzukehren, und die Chance des Verlierens an Arbeiten wegen dem Eintreten von Operationsfehlern, Computer Fehlern und Stromausfällen zu verhindern. Es existieren zwei Wege für Sie von Missgeschicke oder Probleme zurückzukehren:

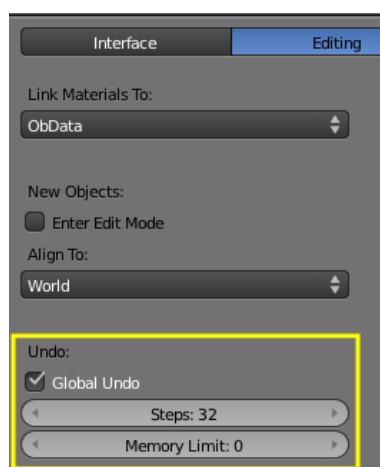
Auf der [Benutzerebene](#) (In Bezug auf Aktionen)

- Für Ihre Aktionen, existieren Optionen wie Rückgängig, Wiederherstellen und einem Verlauf rückgängig machen, zum Zurückkehren von Missgeschicken unter normalen Bedingungen, oder zum Zurückkehren zu einer speziellen Aktion.
- Blender enthält auch neue Funktionen wie Wiederholen und Letztes wiederholen, und dem neuen Letztes wiederholen welches Sie in Verbindung mit den aufgelisteten Optionen nutzen können.

Auf der [Systemebene](#) (In Bezug auf Dateien)

- Es existieren Optionen wie die Automatische Sicherung die Ihre Datei nach einiger Zeit automatisch speichern, und Beim Beenden speichern, welches Ihre Blender Datei automatisch speichert wenn Sie Blender beenden. Anmerkung: In Bezug darauf, dass die anderen Funktionen standardmäßig aktiviert sind, kann die Funktion Beim Beenden speichern nicht deaktiviert werden.

Optionen für Aktionen (Benutzerebene)



Undo options

Die unteren Kommandos erlauben Ihnen versehentliche Aktionen zurückzunehmen, Ihre letzte Aktion zu wiederholen, oder lässt Sie wählen zu einem bestimmten Punkt zurückzuspringen, durch das wählen aus einer Liste von zuletzt verwendeten Aktionen die von Blender mitgeschnitten werden. Zwei neue Funktionen die zur Blender 2.5x Serie hinzugefügt worden sind, sind die Kommandos Wiederholen und Letztes wiederholen.

Zum Aktivieren, oder deaktivieren vom Verlauf der Ihre Aktionen speichert, gehen Sie zum Fenster Benutzereinstellungen und klicken Sie dort auf den Reiter Editieren. In diesem Bereich können Sie folgendes setzen:

Globales Rückgängigmachen

Dies erlaubt Blender Aktionen zu speichern wenn Sie sich **nicht** im Bearbeitungsmodus befindet. Zum Beispiel, Duplizierung von Objekten, das Ändern von Paneleigenschaften oder das Wechseln zwischen Modi. Bei der Standard Blender Installation ist *Globales Rückgängigmachen* standardmäßig aktiviert.

Schritte

Dieses numerische Feld gibt Auskunft darüber wie viele Schritte oder Aktionen gespeichert werden sollen. Der Standardwert von **32** erlaubt Ihnen die zweiunddreißig letzten Aktionen rückgängig zu machen. Sie können dieses numerische Feld auf ein Maximum von **64** einstellen.

Speichergrenze

Dieses numerische Feld erlaubt Ihnen die maximale Anzahl an Speicher in Megabyte anzugeben, die für den Verlauf Ihrer Aktionen genutzt wird. Der Standardwert **0** steht für keine festgelegte Grenze.

Rückgängig

Tastenkombination: StrgZ

Wie in den meisten Programmen, drücken Sie zum rückgängig machen Ihrer letzten Aktion, einfach StrgZ

Wiederherstellen

Tastenkombination: ⇧ ShiftStrgZ

Zum Zurückrollen Ihrer Rückgängig machen Aktion, drücken Sie ⇧ ShiftStrgZ

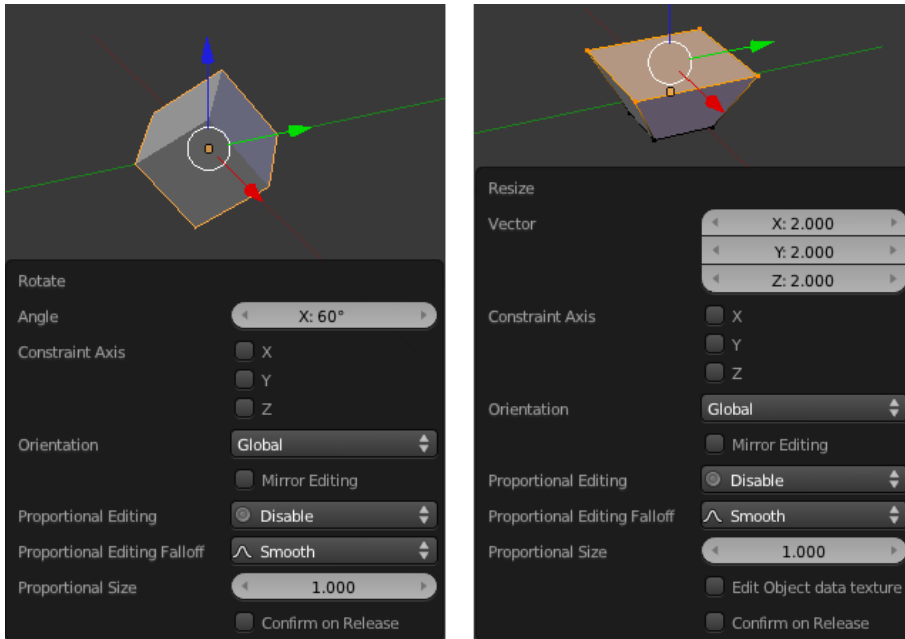
Letztes wiederholen

Tastenkombination: F6

Letztes wiederholen ist das Wiederholen Ihrer letzten Aktion. F6 nach einer Aktion liefert Ihnen ein kontext-basiertes Pop-Up Fenster basierend auf Ihrer zuletzt getätigten Aktion, und dem Modus und Fenster in welchem Blender benutzt wird.

Wenn Ihre letzte Aktion zum Beispiel eine Rotation im Objektmodus war, wird Ihnen das Fenster den zuletzt veränderten Wert anzeigen der für den Winkel verändert wurde, (siehe Bild bei Letztes Wiederholen: Rotieren). Durch das Tippen von **0** (Null) in das numerische Feld können Sie Ihre Aktion auch komplett zurücknehmen. Es gibt noch einige nützliche Optionen, basierend auf dem Kontext Ihrer Aktion, und Sie können Aktionen nicht nur zurücknehmen, sondern auch komplett mittels der verfügbaren Optionen verändern.

Wenn Sie sich im Bearbeitungsmodus befinden, wird das Fenster seinen Inhalt auch basierend auf Ihrer zuletzt getätigten Aktion hin ändern. In unserem zweiten Beispiel (rechts), war die letzte Aktion eine Vertex Verschiebung, wir nahmen die Skalierung einer Fläche vor, und, wie Sie sehen können, sind die Inhalte des Pop-Up Fensters anders, wegen Ihrem Kontext (Bearbeitungsmodus). (siehe Bild bei Letztes Wiederholen: Skalieren)



Letztes wiederholen: Rotieren (Objekt Modus, 60 Grad) _____ Skalieren (Bearbeitungsmodus, Feldskalierung)



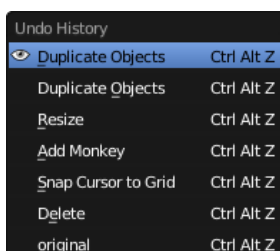
Operationen mit Letztes wiederholen

Einige Operationen erzeugen besonders nützliche Ergebnisse wenn Sie ihre Parameter mit dem F6 Menü ausprobieren. Zum Beispiel, dass hinzufügen eines Kreises. Wenn Sie die Vertex Anzahl auf 3 reduzieren, erhalten Sie ein perfekt gleichseitiges Dreieck

Verlauf rückgängig machen

Modus: Alle Modi

Tastenkombination: CtrlAltZ



Menü zu Verlauf rückgängig machen, welches mit CtrlAltZ erscheint.

Es existiert auch ein Menü zum rückgängig machen Ihres Verlaufes, aufgenommen von Blender. Sie können auf diesen Verlauf mittels CtrlAltZ zugreifen.

Die Funktion zum *Verlauf rückgängig machen* wird Sie zurück zu der Aktion bringen die Sie wählen. Ähnlich zu dem rückgängig

machen Ihrer letzten Aktion mit CtrlZ und zum Wiederholen ⇧ ShiftCtrlZ, können Sie in der Zeitleiste mit "Verlauf rückgängig machen" so lange wie sie möchten herumspringen, so lange Sie keine Änderung durchführen. Sobald Sie eine neue Änderung durchführen, führt dass zu einem neuen Verlauf, und Ihr alter Verlauf geht verloren.

Letztes wiederholen

Modus: Alle Modi

Tastenkombination: ⇧ ShiftR

Die Funktion Letztes Wiederholen wird Ihre letzte Aktion wiederholen, indem Sie ⇧ ShiftR drücken.

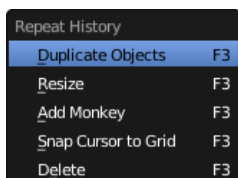
In den unteren Beispielbildern, haben wir ein *Affen* Mesh dupliziert, und das Objekt ein bisschen bewegt. Mit Letztes wiederholen ⇧ ShiftR, wurde der *Affe* wieder dupliziert und bewegt.



Letztes wiederholen

Modus: Alle Modi

Tastenkombination: F3

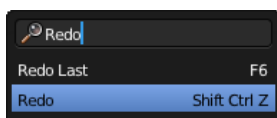


Letztes wiederholen, welches mit F3 erscheint.

Letztes wiederholen wird Ihnen eine Liste der zuletzt wiederholten Aktionen anzeigen, woraus Sie dann wählen können, welche Sie wiederholt haben möchten. Es funktioniert auf die selbe Weise wie Verlauf rückgängig machen, oben beschrieben, aber die Liste enthält nur Aktion zum Wiederholen. Um Letztes wiederholen aufzurufen, drücken Sie F3.

Es existierten zwei verschiedene Verläufe für Blender
Blender nutzt zwei separate Verläufe, einen für den Bearbeitungs Modus, und einen für den Objekt Modus.

Blender Suche



Spacebar search for Redo
Last

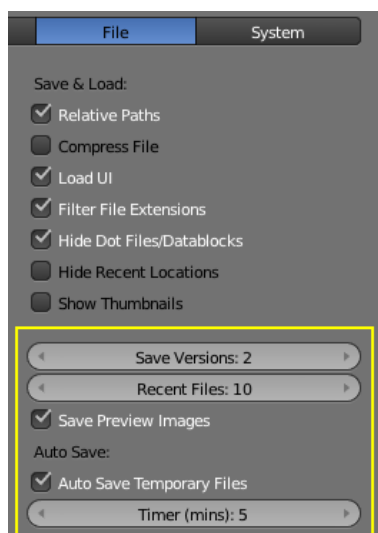
Sie können jederzeit auf alle erklärten Optionen für Benutzeraktionen mittels der Blender Suche mit Leertaste zugreifen.

Wichtiger Hinweis

Wenn Sie Blender verlassen, geht die komplette Liste der Benutzeraktionen verloren, sogar falls Sie Ihre Datei vor dem Beenden gespeichert haben.

Optionen für Dateien (Systemebene)

Speichern und Automatische Sicherung



Automatische Sicherungsoptionen

Der Rechner stürzt ab, Stromausfall oder einfach vergessen zu speichern kann zu Verlusten oder Beschädigungen Ihrer Arbeit führen. Um die Chance von Dateiverlust wenn diese Ereignisse auftreten zu reduzieren, kann Blender Funktionen zur Automatischen Sicherung verwenden. Der Reiter Datei des Fensters Benutzereinstellungen erlaubt Ihnen zwei Wege zu konfigurieren die Blender Ihnen bereitstellt um die Unterschiede zur letzten Version Ihrer Arbeit zu minimieren.

Beim Beenden speichern

Die Funktion Beim Beenden speichern ist standardmäßig in Blender aktiviert. Blender wird immer Ihre Dateien speichern wenn Sie die Anwendung unter normalem Gebrauch beenden.

Save Versions

This option tells Blender to keep the indicated number of saved versions of your file in your current working directory when you manually save a file. These files will have the extension: `.blend1`, `.blend2`, etc., with the number increasing to the number of versions you specify. Older files will be named with a higher number. e.g. With the default setting of **2**, you will have three versions of your file: `*.blend` (your last save), `*.blend1` (your second last save) and `*.blend2` (your third last save).

Temporäre Dateien automatisch sichern

Checking this box tells Blender to *automatically* save a backup copy of your work-in-progress to the Temp directory (refer to the File panel in the User Preferences window for its location). This will also enable the Timer(mins) control which specifies the number of minutes between each Auto Save. The default value of the Blender installation is **5** (5 minutes). The minimum is **1**, and the Maximum is **60** (Save at every one hour). The Auto Saved files are named using a random number and have a `.blend` extension.



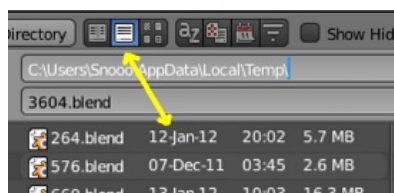
Compress Files

The option to Compress files will try to compact your files whenever Blender is saving them. Large Scenes, dense Meshes, big Textures or lots of elements in your Scene will result in a big `.blend` being created. This option could slow down Blender when you quit, or under normal operation when Blender is saving your backup files. In fact, using this option you will trade processor time for file space.

Automatische Sicherungen wiederherstellen

Letzte Sitzung wiederherstellen

Datei » Letzte Sitzung wiederherstellen... wird die `quit.blend` öffnen die in Ihrem Temp Verzeichnis gespeichert ist wenn Sie Blender beendet. Beachten Sie dass Dateien in Ihrem Temp bei einem Neustart automatisch gelöscht werden.



Blender Dateiverwaltung

- Ein Tipp: Wenn Dateien wiederhergestellt werden, wechseln Sie zu Ihrem temporären Verzeichnis. Es ist wichtig, während der Ordnerdurchsuchung, die detaillierte Listenansicht zu aktivieren. Andernfalls, werden Sie nicht in der Lage sein die Datumsangaben der automatisch-gesicherten `.blend` Dateien zu ermitteln. (Siehe Bild: Blender Dateiverwaltung)

Automatische Sicherung wiederherstellen

Datei » Automatische Sicherung wiederherstellen... erlaubt Ihnen die automatisch gespeicherte Datei zu öffnen. Nachdem die automatisch gespeicherte Version geladen ist, können Sie sie über die momentane Datei in Ihrem Arbeitsverzeichnis als eine

gewöhnliche `.blend` Datei speichern.

Wichtiger Hinweis

Wenn Sie eine Automatische Sicherung wiederherstellen, werden Sie alle Änderungen bis zum letzter Automatischen Sicherung verlieren.

Nur **eine** automatisch gesicherte Datei existiert für jedes Projekt (z.B behält Blender keine älteren Versionen – was heißt dass sie nicht weiter als ein paar Minuten zurück mit diesem Werkzeug kommen werden).

Andere Optionen

Zuletzt geöffnete Dateien

Diese Einstellung kontrolliert wie viele zuletzt geöffneten Dateien im Unter-Menü Datei » Andere Optionen aufgelistet werden.

Vorschaubilder speichern

Vorschaubilder von Bildern und Material in der Dateiverwaltung welche je nach Bedarf automatisch generiert werden. Um diese Vorschaubilder in Ihre `.blend` Datei zu speichern, aktivieren Sie diese Option (mit dem Preis Ihre `.blend` Datei zu vergrößern).

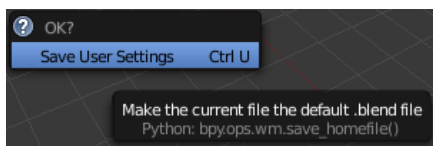
Setting the default Scene

Modus: Alle Modi

Tastenkombination: CtrlU

Menü: Datei » Benutzereinstellungen speichern

Wenn Sie Blender starten oder ein neues Projekt mit dem Menü Eintrag Datei » Neu oder durch die Tastenkombination StrgN anlegen, wird eine neue Szene aus der Standard Szene in den Benutzereinstellungen erstellt.





Save User Settings popup

To change the default scene, make all of the desired changes to the current scene and press CtrlU.

The Save User Settings popup confirmation will appear. Click LMB  on the Save User Settings popup or press ↵ Enter.

Press Esc to abort.

Vom Menü aus

To change the default Scene from the menu, make all of the desired changes to the current Scene you are working on and LMB  in File » User Preferences. The User Preferences Editor Window will appear. In the Interface Tab, click LMB  on the Save As Default button to save the changes. The current scene, all objects, and settings will then be saved in the User Preferences as your starting Scene. (See Fig: Image of the User Preferences Window, with *Save as Default* highlighted in yellow).

If you don't want to use your Scene and/or Blender customization as your default Scene, leave the User Preferences Editor Window without clicking the Save As Default button, and your default Scene when starting Blender will remain unchanged.

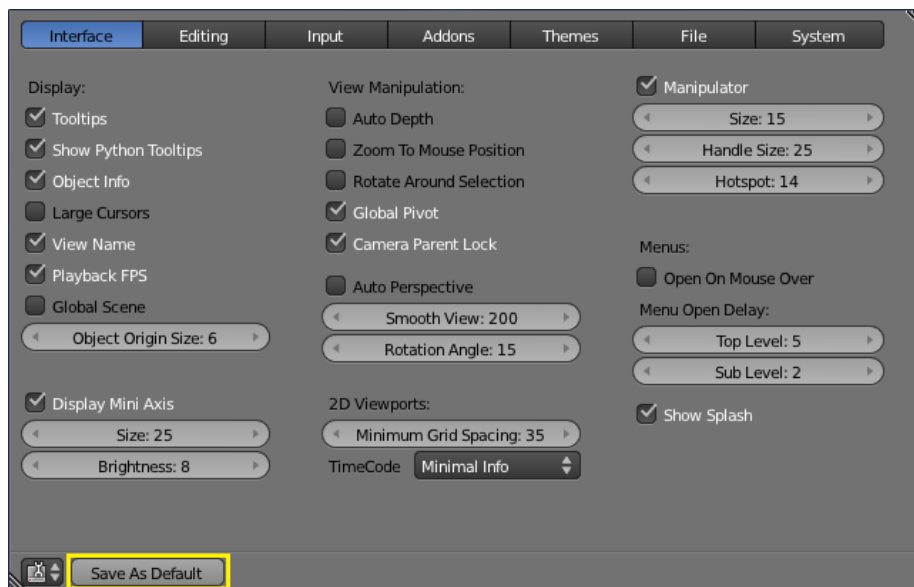



Image of the User Preferences Window, with *Save as Default* highlighted in yellow

Zurücksetzen der Standard Szene auf Werkseinstellungen

Modus: Alle Modi

Tastenkombination: undefiniert, you can add one for your [Keymap](#) »

Menü: Datei » Load factory Settings

To restore the default scene to the factory settings, LMB  in File » Load Factory Settings. This will restore all User Preferences back to the original Factory Settings. To save the changes, use CtrlU and your Factory Settings will be saved as the default Scene for Blender.

User Preferences Window

For more information about the Editor Window for User Preferences or how to clean your preferences manually, please read the chapter about [User Preferences](#)

Bildschirmfotos und -aufzeichnungen

Um die Teamarbeit und Rapid Prototyping zu erleichtern, möchten Sie vielleicht Bilder schießen oder Videos Ihrer Arbeit erstellen. In diesem Kapitel erklären wir die verfügbaren Optionen in Blender, und am Ende der Seite weisen wir auf einige nützliche Tastenkombinationen zum Aufnehmen von Bildschirmfotos für die am meisten genutzten Betriebssystemen sowie eine Auswahl von optionaler Software hin.

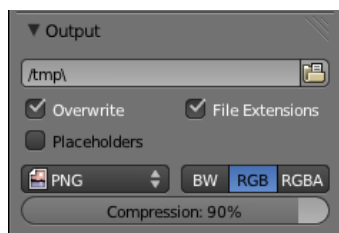
In Blender haben Sie die folgenden Optionen:

- Blender Bildschirmfotos (Screenshots): Sehr nützlich zum Veranschaulichen gegenwärtiger Arbeitsschritte oder für Hilfeanfragen, dem Zeigen von verschiedenen Stufen der Modellierung oder zur Hervorhebung wichtiger Details Ihrer Arbeit.
- Blender Bildschirmaufzeichnungen (Screencasts): Nützlich zum Lernen, zum Darstellen von Arbeitsschritten in Zeitraffer oder auch für Testvideos sowie dem Abspielen manuell erstellter Bewegungsphasen in zeitlicher Abfolge.

Blender-Bildschirmfotos (Screenshots)

Modus: Alle Modi

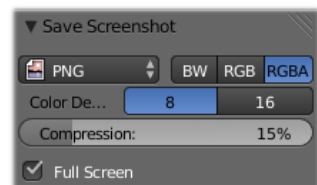
Tastenkombination: StrgF3



Ausgabepanel PNG-Bildformat

Die Tastenkombination StrgF3 nimmt ein Bildschirmfoto des Blender-Arbeitsfensters auf und öffnet daraufhin das Fenster zur Blender-Dateiverwaltung, die es Ihnen erlaubt, einen Namen für ihr Bildschirmfoto sowie den Speicherort auszuwählen. Im Beispielfoto rechts wird das PNG-Format als Ausgabeformat für das Bildschirmfoto verwendet. Das Bildschirmfoto wird in dem Format gespeichert, das im [Ausgabepanel](#) des [Render-Kontext](#)-Fensters festgelegt worden ist (Siehe Abbildung: Ausgabepanel PNG-Bildformat).

- Blender-Bildschirmfotos werden mit derjenigen vollen Breite und Höhe des Blender-Arbeitsfensters aufgenommen, das Sie im Moment des Befehlsaufrufs benutzen.



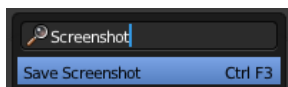
Option zum Speichern von Bildschirmfotos

Es besteht auch die Option, nur das Fenster aufzunehmen, in dem Sie gerade arbeiten. Wenn sich für Sie das Fenster der Blender-Dateiverwaltung öffnet, befindet sich links der Reiter *Save Screenshot* (Bildschirmfoto speichern), auf dem Sie ein Kontrollkästchen mit der Option *Full Screen* (Vollbildschirm) finden (Siehe Bild: Bildschirmfoto speichern Option).

- Aktivieren Sie die Option zum Aufnehmen des gesamten Blender-Fensters.
- Deaktivieren Sie das Kontrollkästchen, um nur das aktive Fenster (also das, in dem sich Ihre Maus befindet, wenn Sie den Befehl aufrufen) aufzunehmen.



Tastaturkombinationskonflikte



Suchfunktion

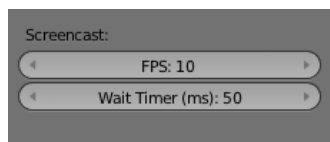
Mitunter kommt es vor, dass Ihr Betriebssystem einige Tastaturkürzel für sich beansprucht, die auch Blender standardmäßig für seine Funktionen vorsieht. In diesem Fall können Sie die in Blender vorhandene Suchfunktion benutzen (Siehe Fig: Search Functionality). Hierzu betätigt man Space und gibt Screenshot in das Suchfenster ein.

Blender-Bildschirmaufzeichnung (Screencasts)

Modus: Alle Modi

Tastenkombination: AltF3

Die Tastenkombination AltF3 startet die Bildschirmaufzeichnungsfunktion. Diese Funktion wird Ihre Bedienhandlungen in zeitlicher Abfolge entweder als Video oder als Sequenz von Bilddateien abspeichern. Typ und Ablageort der Ausgabe werden über die Einstellungen im [Output panel](#) des [Render context](#)-Panels festgelegt. Mit den Grundeinstellungen wird eine Bildschirmaufzeichnung aus einer Serie von PNG-Bilddateien generiert, die alle 50 ms aufgenommen und im /tmp-Ordner abgelegt werden. Möchten Sie ein Video aufnehmen, müssen Sie den Output auf eines der Movie File Formats setzen, die von Ihrem System unterstützt werden und im Output panel-Formatmenü aufgelistet sind. Falls Sie sich unsicher sind, welche Video-Codecs von Ihrem System unterstützt werden, wählen Sie AVI JPEG.



Optionen im User Preferences-Editor

Die Bildfrequenz pro Sekunde (FPS = Frames per Second) sowie die Zeit zwischen jedem Bildschirmfoto einer Bilderserie kann festgelegt werden auf dem [System panel](#) des [User Preferences](#)-Fensters.

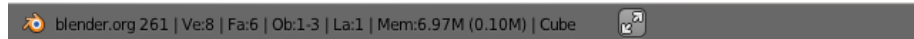
(Siehe Abbildung: Optionen im *User Preferences*-Editor)

Audio-Unterstützung

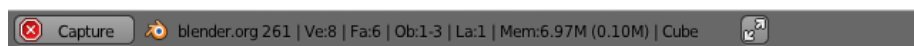
Die Blender-Bildschirmaufzeichnung unterstützt keine Audioaufnahmen, so dass Sie diese mittels anderer Software, z.B. [Audacity](#), in Verbindung mit Blender nachvertonen müssen.

Sobald Sie die Blender-Bildschirmaufzeichnung starten, wird die Header-Menüzeile des Info Windows dahingehend wechseln, dass eine Stop-Taste zum Anhalten der Aufzeichnung eingeblendet wird.

Unten zeigen wir zum einen die normale Header-Menüzeile des Info Windows im normalen Blender-Bearbeitungsmodus (Siehe Abbildung: Info Window - Header - Normaler Blender-Bearbeitungsmodus) und zum anderen die Zeile mit Stop-Taste während des Aufzeichnungsmodus. (Siehe Abbildung: Info Window - Header - Stop-Taste für die Bildschirmaufzeichnung).



Info-Fenster - Header - Normaler Blender-Bearbeitungsmodus

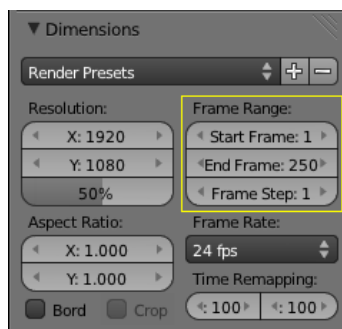


Info-Fenster - Header - Stop-Taste für die Bildschirmaufzeichnung

(Anmerkung: Die Aufnahme der Header-Menüzeile erfolgte in Blender 2.61)

Der einzige Weg zum Stoppen der Bildschirmaufzeichnung

Nur das Betätigen der Stop-Taste hält die Bildschirmaufzeichnung an. Wenn Sie Esc drücken, wird dies lediglich für Bedienhandlungen im Blender-User Interface Auswirkungen zeigen (Anhalten von Animationen, Playbacks usw.), die Screencasts-Funktion wird dadurch jedoch nicht angehalten.



Dimensions-Panel - Frame Range

Die Einzelbilder (*Frames*) werden mit einem Suffix (Dateinamenerweiterung) hinter ihrem eigentlichen Dateinamen gespeichert, wobei sich dieser Suffix aus den Nummern aus den Feldern für *Start Frames* und *End Frames* zusammensetzt, die bei *Frame Range* (Einzelbildanzahl) auf dem Dimensions-Panel des [Render-Kontexts](#) festgelegt werden. (Siehe Abbildung: Dimensions Panel - Frame Range - gelb hervorgehoben)

Wichtig:

Die Einstellung des End-Frames, zu sehen unter Frame Range im Dimension-Panel, **wird nicht** zum automatischen Anhalten der Aufnahme führen. Eine Aufnahme können Sie immer nur von Hand mittels der Stop-Taste beenden.

Die Videos werden intern in derselben Weise wie die Bildschirmfotos generiert, wobei jeweils die aktuelle Breite und Höhe des Arbeitsfenster zugrundegelegt wird. Wenn Sie eine Aufzeichnung in eine Video-Datei auswählen, wird Blender diese Einzelbilder an einen Video-Codec übermitteln.

Warnung: Einige Codecs schränken die Breite/Höhe oder die Videoqualität ein.

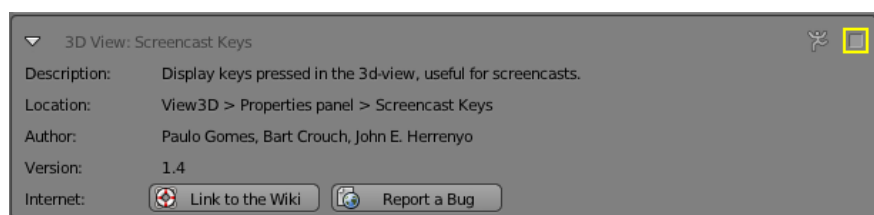
- Speichern Sie Ihre Bildschirmaufzeichnung in einem Bildformat, werden die Blender-Fenster als vollformatige Bilder mit originaler Breite und Höhe abgespeichert, und die Qualität des Bildes wird von dessen Dateiformat (JPG, PNG usw.) und deren Einstellungen (z.B. Schieberegler *quality* für das JPG-Format) abhängig sein.
- Wenn Sie Ihre Bildschirmaufzeichnung in ein Videoformat speichern, wird diese an einen Codec übergeben. In Abhängigkeit von den jeweiligen Codec-Einschränkungen wird das so erzeugte Ausgabevideo möglicherweise herunterskaliert. Des weiteren können bestimmte Kombinationen von Fensterbreite und -höhe von einige Codices nicht verarbeitet werden. In diesen Fällen wird die Bildschirmaufzeichnung versuchen zu starten, aber sogleich wieder anhalten. Um dieses Problem zu lösen, sollten Sie ein anderes Fensterformat und/oder einen andern Codec wählen.

Blender-Fensterdimensionen

Es gibt eine Möglichkeit, die Abmaße des Blender-Arbeitsfensters mit denen der auszugebenden Videodatei dergestalt in Übereinstimmung zu bringen, dass die Blender-Bildschirmaufzeichnung gängigen Standard-Dimensionen entspricht (NTSC, HD, Full HD usw.). Man kann die Breite und Höhe des Blender-Arbeitsfensters durch einen Start von Blender über eine Kommandozeile steuern. Um mehr zum Start von Blender über die Kommandozeile zu erfahren, lesen Sie bitte nach unter [Blender-Konsolenfenster](#).

Addon: 3D View: Screencast Keys (Tastenanzeige)

Das aus der Community hervorgegangene Addon 3D View:Screencast Keys zeigt Ihnen in der linken unteren Ecke des 3D-Ansichtsfensters die Tasten, Tastenkombinationen und Mausbedienungen, sobald diese während einer Bildschirmaufzeichnung benutzt werden. Dieses Addon der Community wird mit der Standardinstallation von Blender ausgeliefert. Untenstehendes Bild zeigt dieses Community-Addon mit seinem geöffneten Tab (Siehe Abbildung: 3D View: Screencast Keys - Addon). Um dieses Addon zu aktivieren, öffnen Sie das Bearbeitungsfenster für die User Preferences CtrlAltU, gehen auf den Tab Addons und dort zu den Addons für *3D View*. Zum Aktivieren des Addons einfach auf das Kästchen klicken (gelb markiert).

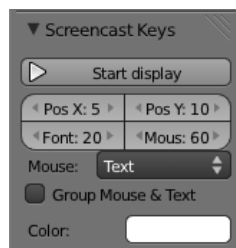


3D View: Screencast Keys - Addon

Mode: Alle Modi → Addon Enabled

Hotkey: N zum Anzeigen des Properties Panel → Screencast Keys Tab

Menu: View » Properties → Screencast Keys Tab



Screencast Keys
Addon Tab - Properties
Panel

Sobald das Addon installiert worden ist, werden Sie das folgende Tastenfeld für die Bildschirmaufzeichnung am Ende der Liste auf dem Properties Panel sehen.

Beschreibung:

- **Start display button:** Sobald Sie diese Taste gedrückt haben, wird Blender jede Taste oder Tastenkombination in der unteren linken Ecke des 3D-Bearbeitungsfensters als Fließtext sehen können. Betätigen Sie dieselbe Taste oder Tastenkombination mehrfach, wird Blender eine "xn"-Kennzeichnung am Ende der Tasten bzw. Tastenkombination hinzufügen, wodurch die Anzahl der Betätigungen einer oder mehrerer Tasten angezeigt wird.
- **Stop display button:** sorgt dafür, dass Blender keine Tasten mehr in die Bildschirmaufzeichnung einblendet.

- **PosX:** Position der Tastendarstellung entlang der **X**-Achse.
- **PosY:** Position der Tastendarstellung entlang der **Y**-Achse.
- **Font:** Schriftart.
- **Mouse:** Größe des Mauszeigers.
- **Mouse display:** In diesem Drop-Down-Menü können Sie einstellen, wie der Text in der Bildschirmaufzeichnung dargestellt werden soll.
- **Text:** Darstellung der betätigten Tasten und Maustasten als Text.
- **Icon:** Darstellung der Maus als Icon und der gedrückten Tasten als Text.
- **None:** Ausschließliche Darstellung betätigter Tasten ohne Maus-Informationen.
- **Group Mouse & Text Check box:** Bei Auswahl dieser Option wird Blender einen Kasten um die eingeblendeten Texte erzeugen, um die Lesbarkeit zu erleichtern.
- **Color:** Auswahl der Textfarbe.



Neues Community-Addon

Inzwischen gibt es auch ein Addon für Blender 2.5/2.6, womit sich Bildschirmfotos von jeder beliebigen Stelle machen und sogleich direkt nach [Pasteall](#) hochladen lassen. Dieses Addon besitzt gegenwärtig keine Entwicklerseite, wird aber hierher verlinkt werden, sobald es fertig entwickelt sein wird.

Aufnahme von Bildschirmfotos mittels Betriebssystem

Sie können auch das Betriebssystem zum Aufnehmen des Bildschirms in die Zwischenablage verwenden. Anschließend können Sie das Bild aus der Zwischenablage in Ihren Grafik-Editor einfügen.

Windows-Bildschirmfotos

Drücken Sie Alt+Druck zum Aufnehmen des aktiven Programmfensters in die Zwischenablage.

Mac OSX-Bildschirmfotos

Drücken Sie \mathbb{A} Cmd+Shift3 zum Aufnehmen des Bildschirms in eine Datei auf dem Schreibtisch.

Drücken Sie Ctrl+ \mathbb{A} Cmd+Shift3 zum Aufnehmen des Bildschirms in die Zwischenablage.

Drücken Sie \mathbb{A} Cmd+Shift4 zum Aufnehmen eines Bildschirm-Bereiches in eine Datei auf dem Schreibtisch.

Drücken Sie Ctrl+ \mathbb{A} Cmd+Shift4 zum Aufnehmen eines Bildschirm-Bereiches in die Zwischenablage.

GNU/Linux-Bildschirmfotos

Bei einigen Linux-Distributionen (wie Ubuntu) und Fenstermanagern können Sie die Druck-Taste drücken, um ein Bildschirmfoto in eine Datei zu speichern. Für andere Distributionen oder Fenstermanager benötigen Sie möglicherweise zusätzliche Programme. Beispiele dafür sind hierfür ohne Anspruch auf Vollständigkeit: [xvidcap](#), [scrot](#) und [recordMyDesktop](#). Für weitergehende Informationen greifen Sie bitte auf Ihr Distributionshandbuch oder Software Repository zurück.

Bildschirmfotos mittels Software-Tools

Ergänzend zu den Optionen, die Ihnen Blender und Ihr Betriebssystem zum Aufnehmen von Bildschirmfotos bieten, gibt es weitere hilfreiche Programme zum Aufnehmen von Bildschirmfotos, beispielsweise GIMP, Photoshop, Screenhunter usw.

GIMP-Bildschirmfotos

Aufnehmen von Bildschirmfotos mit GIMP:

- Gehen Sie zu Datei -> Erstellen -> Bildschirmfoto.
- Es gibt zwei Optionen:
 - Aufnehmen eines Bildschirmfotos für ein bestimmtes Fenster
 - Aufnehmen eines Bildschirmfotos für den ganzen Bildschirm

Darüber hinaus gibt es ein Feld, in das man die Verzögerungszeit bis zur folgenden Aufnahme in Sekunden eintragen kann. Wählen Sie die gewünschten Optionen und Drücken Sie auf die Taste *Auslösen*. Wenn Sie sich dafür entscheiden, ein Bildschirmfoto eines einzelnen Fensters aufzunehmen, müssen Sie nach dem Ende der verspäteten Auslösung auf ein Fenster klicken.

Das Hilfe System

Modus: Alle Modi

Tastenkombination: Undefined - You can add one for your [Keymap »](#)

Menü: Help

Blender stellt eine Reihe an eingebauten und webbasierten Hilfe Optionen bereit.

Die eingebauten Hilfe Optionen beinhalten:

- Ein Menü mit allen Hilfe Optionen, welches die webbasierten beinhaltet. Einige von ihnen sind auch auf dem Willkommensbildschirm zu finden.

Andere neue Funktionen wie:

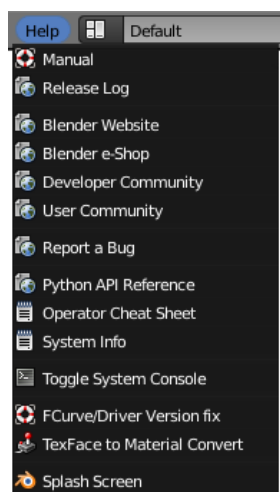
- Die Blender Suche (neue Funktion).
- Tooltips die die internen Python Operatoren anzeigen (neue Funktion), wenn der Benutzer mit der Maus über einen Knopf, ein Menü, ein numerisches Feld oder jede andere Blender Funktion fährt die über eine benanntes Python Operatorol verfügt.

Allgemeine Webbasierte Hilfe Optionen



Browser und Internet Verbindung

Einige Formen an Hilfe starten Ihren Webbrowser und greifen auf die Webserver der Blender Foundation zu. Um dies zu tun müssen Sie über einen konfigurierten Webbrowser für Ihr Betriebssystem, und eine Verbindung mit dem Internet haben.



Hilfe Menü

- [Manual](#) - Dies ist ein Verweis zum offiziellen Blender Handbuch, im

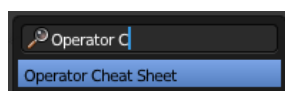
Wiki Format, welches Sie momentan lesen.

- [Release Log](#) - Die Veröffentlichungshinweise zur momentanen Blender Version.
- [Blender Website](#) - Die blender.org Homepage.
- [Blender e-Shop](#) - Der Blender e-Store, wo Sie Training DVD's, Bücher, T-Shirts und andere Produkte kaufen können.
- [Developer Community](#) - Die blender.org "Get Involved" Seite. Das ist die Startseite für die Blender Software Entwicklung, Fehler Sammlung, Patches und Skripte, Bildung und Training, Dokumentationen über die Entwicklung und Funktionalitäts Forschung.
- [User Community](#) - Liste von vielen verschiedenen Hilfestellen
- [Report a Bug](#) - Die Blender Fehlermelde Stelle

Wichtig: Um einen Fehler zu melden, müssen Sie auf der Webseite registriert sein.

Programmier Optionen

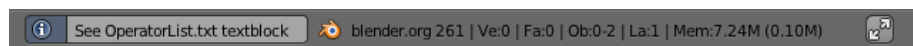
- [Python 2.6X API Referenz](#) - Das Python Application Programming Interface (API) das Blender und [Python](#) zum kommunizieren unter einander benutzen. Nützlich für die Blender Game Enginee, Anpassung, und anderes Skripting.



Operator Cheat Sheet

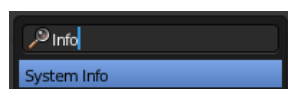
- Operator Cheat Sheet - Erstellt die `OperatorList.txt` Datei, auf welche Sie im Text Editor zugreifen können. Sie können auch die Blender Suche nutzen um diese Datei zu generieren. The text will list the available Python operators. At the time we were writing this part of the Manual (Blender 2.61), Blender had 1245 Operators.

While Blender is generating this list, the Info Window will change, showing a message for the operation (See Fig: Info Window – Operator Cheat Sheet). To read the Text, switch to the Blender Text Editor Window, using the [Window type Selector](#), and then, clicking on the button *Browse Text to be Linked* of the Text Editor, your text block will be shown in the Editor. The file will be in your list of Text files, named as *OperatorsList.txt*, if the file is already generated, Blender will add a numeric suffix for the subsequent ones.



Info Window – Operator Cheat Sheet

Optionen zum Diagnostizieren



Blender Search - System Info

- System Info - Creates a `system-info` file, which you can access in the Blender Text Editor. The text lists various key properties of your system and Blender, which can be useful in diagnosing problems. When you click on this Option, Blender will verify your installation, will change the Info Window for a while when generating the file (See: Info Window – Info.txt). You can also use Blender Search to generate the file.

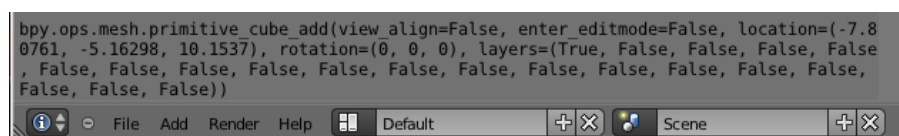
To read the Text, switch to the Blender Text Editor Window, using the [Window type Selector](#), and then, clicking on the button *Browse Text to be Linked* of the Text Editor, your text block will be shown in the Editor. The file will be in your list of Text files, named as *system-info.txt*, if the file is already generated, Blender will add a numeric suffix for the subsequent ones.

- The text file is created with **4** different sections: Blender, Python, Directories and OpenGL, which we will explain below:
 - **Blender:** This section of the info.txt shows you the Blender version, flags used when Blender was compiled, day and time when Blender was compiled, build system, and the path in which Blender is running.
 - **Python:** The Python version you are using, showing the paths of the Python programming language paths.
 - **Directories:** The Blender directories setup for `scripts`, `user scripts`, `datafiles`, `config`, `scripts (internal)`, `autosave` directory and `temp dir`. Those directories are configured using the [User Preferences](#) Editor Window.
 - **OpenGL:** This section will show you the version of OpenGL that you are using for Blender, the name of the manufacturer, version, vendor and a list with your card capabilities or OpenGL software capabilities.



Info Window – Info.txt

- Toggle System Console - Reveals the command window that contains Blender's stdout messages. Can be very useful for figuring out how the UI works, or what is going wrong if you encounter a problem. Even more information is available here, if you invoke Blender as `blender -d`. This menu item only shows up on Windows.
 - In all Operating Systems, to see this information, simply run `blender` from the command-line.
 - On Linux, if you ran Blender from the GUI, you can see the output in `~/.xsession-errors`
 - On Mac OS X, you can open `Console.app` (in the Utilities folder in Applications) and check the Log there.
- Info Window Log - This is not exactly a Help menu, but it is related. If you mouseover the line between the Info window and the 3D then click and drag the Info window down a bit, you can see the stream of Python calls that the UI is making when you work. This can be useful in creating scripts.

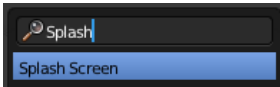


The Info Window Log after adding a Cube

Legacy Version Support

- FCurve/Driver fix - Sometimes, when you load .blend's made from older versions of Blender (2.56 and previous), the Function Curves and Shapekey Drivers will not function correctly due to updates in the animation system. Selecting this option updates the FCurve/Driver data paths.
- TexFace to Material Convert - Convert old Texface settings into material. It may create new materials if needed.

Splash Screen



Splash Screen Search

Splash Screen - This displays the image where you can identify package and version. At the top-right corner, you can see the Version and SVN (Subversion) revision (See Fig: Blender Splash Screen). For example, in our Splash Screen, you can see the version **2.61.0** and the revision number **r42615** highlighted in green. This can be useful to give to support personnel when diagnosing a problem. You can also use Blender Search to Show the Splash Screen or click in the Small Blender Logo present in the Info Window

There are some Internet Based Help options that are also present in the Blender Splash Screen, those options are highlighted in Yellow in the image.



Blender Splash Screen, Blender Version 2.61

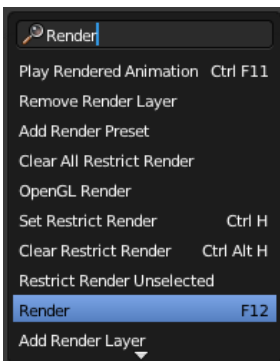
Other Help Options

Here we explain the two new features added for Blender, Blender Search and the recoded Tooltips.

Blender Search

Mode: All modes

Hotkey: Space



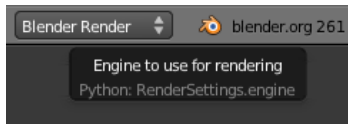
Blender Search - Render

The Blender Search feature, called Blender Search, is a new functionality added by the Blender recode (from 2.4x series to 2.5x series and so on). The Internal name of the feature is *Operator Search*. When you hit Space from your keyboard, Blender will present you with a small Pop Up Window, no matter which Blender Window your Mouse pointer is located (except the Text Editor Window and Python console), and a field for you to type in. Just type what you need and Blender will present you a list of available options. You can click on the appropriate function for you, or search through them using your keyboard, type `↵` Enter to accept, or `Esc` to leave. Clicking outside of the Blender Search Window or taking the Mouse pointer away, will also leave Blender Search.

The Image at the right shows Blender Search when we type the word *Render* inside the field. If you continue typing, your search keywords will refine your search and if no named operator can be found, the small Pop Up Window for the Blender Search will stay blank.

- How it works:
 - Every Blender Internal Operator can use a defined name, some of them are predefined names for the user. For example, the Render command is a named Python call, the appropriate Operator is `Python: bpy.ops.render.render()`, but for the user, it is called Render. All of those *user* names that were previously attributed for Python operators can be searched for using Blender Search.

Tooltips

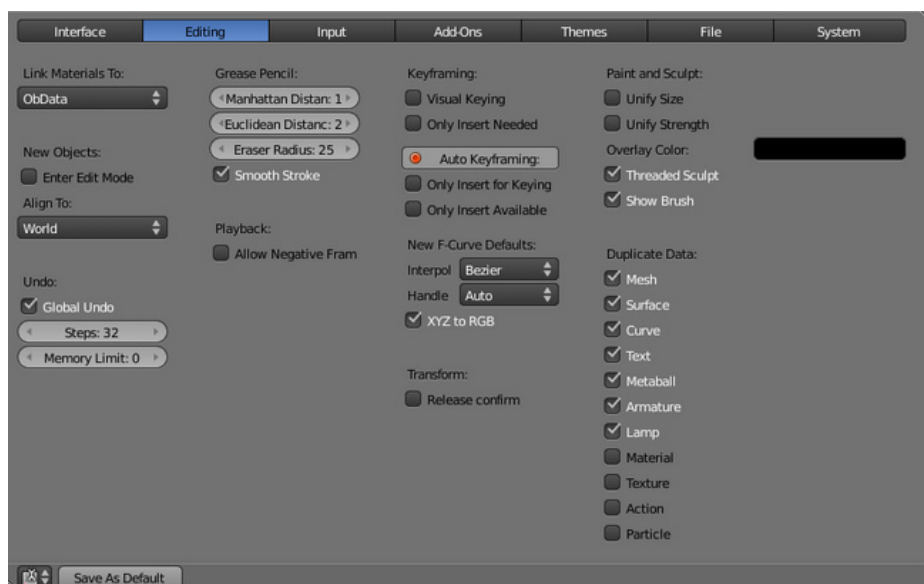


The Mouse pointer was Stopped for a while over the Render Engines List in the Info Window. The normal Tooltip is in white and the Python operator is showed in grey

The Tooltips in Blender were completely recoded, and every time you hover your Mouse over a Button, a Command, Numeric Fields or things that are related to Operators, staying for a while, it will show you not only the normal Tooltip, but also the specific related operator. Those operators are useful for lots of tasks, from Python Scripts to Keymaps. In the example Image at the right, we pointed our Mouse over the Info Window, specifically over the list of the Render engines available, waited for a while, and the Tooltip with the appropriate operator was shown. In our example, it shows the Tooltip *Engine to Use for Rendering* in white, and `Python: RenderSettings.engine` in grey, which is the Operator associated with the function.

Open User Preferences

To open a Blender User Preferences editor go to File » User Preferences or press **CtrlAltU**. Mac users can press **⌘ Cmd.** You can also load the Preferences editor in any window by selecting  User Preferences from the Window type selection menu.



This editor permits you to configure how Blender will work. The available options are grouped into seven tabs, accessible at the top of the window. The options are: *Interface*, *Editing*, *Input*, *Add-Ons*, *Themes*, *File* and *System*.

Configure

Now that you have opened the User Preferences editor, you can configure Blender to your liking. Select what you want to change in the following list:

[Interface](#) • [Editing](#) • [Input](#) • [Add-Ons](#) • [Themes](#) • [File](#) • [System](#)

Save the new preferences

Once you have set your preferences, you will need to manually save them otherwise the new configuration will be lost after a restart or when starting a new scene. Blender saves its preferences with each scene which can be convenient when you need a certain layout or special addons.

On the header (in this case a footer) of the User Preferences window, click on Save As Default. This will save:

- All of the new preferences.
- The currently open scene as the default scene.

You can also save the User Preferences at any time by pressing **CtrlU**.

Load Factory Settings

There are two ways to restore the default Blender settings:

1. Go to File » Load Factory Settings and then save the preferences with **CtrlU** or via the User Preferences editor.
2. Delete the `startup.blend` file from the following location on your computer:
 - Linux: `/home/$user/.blender/'Version Number'/config/startup.blend` (you'll need to show hidden files).
 - Windows 7 and Windows Vista: `C:\Users\%user%\AppData\Roaming\Blender Foundation\Blender\'Version Number'\config\startup.blend`
 - MacOS: `/Users/$user/Library/Application Support/Blender/'Version Number'/config/startup.blend` (you'll need to show hidden files).

While you're in the Blender config folder, it can be valuable to copy your Blender settings file to another folder. In the event that you lose your configuration, you can restore your Blender settings file with your backup copy.

Oberfläche

Tooltips

Wenn aktiviert, erscheint ein Tooltip, ein kurzer Beschreibungstext, wenn sich Ihr Mauszeiger über einem Kontrollknopf befindet. Dieser Hinweis informiert Sie über die Funktion der Sache die unter dem Mauszeiger liegt, zeigt den zugewiesenen Tastenkürzel an (wenn vorhanden) und die Python Funktion die darauf verweist.

Objektinformation

Zeigt den aktiven Objektnamen und die Frame Nummer an der unteren linken Seite der 3D Ansicht an.

Large Cursors

Use large mouse cursors when available.

View Name

Display the name and type of the current view in the top left corner of the 3D window. For example: User Persp or Top Ortho.

Playback FPS

Show the frames per second screen refresh rate while an animation is played back. It appears in one of the viewport corners.

Global Scene

Zwingt die momentane Szene in allen anderen Szenen angezeigt zu werden (ein Projekt kann aus mehr als einer Szene bestehen).

Object Origin Size

Diameter of 3D Object centers in the view port (value in pixels from 4 to 10).

Display Mini Axis

Show the mini axis at the bottom left of the viewport.

Size

Size of the mini axis.

Brightness

Adjust brightness of the mini axis.

Properties Window

Width Check

When the Properties window has a width below this value, it will display its buttons in one column rather than two.

View manipulation

Auto Depth

Use the depth under the mouse to improve view pan/rotate/zoom functionality.

Zoom to Mouse Position

When enabled, the mouse pointer position becomes the focus point of zooming instead of the 2D window center. Helpful to avoid panning if you are frequently zooming in and out.

Rotate Around Selection

The selected object becomes the rotation center of the viewport.

Global Pivot

Lock the same rotation/scaling pivot in all 3D views.

Auto Perspective

Automatically to perspective Top/Side/Front view after using User Orthographic. When disabled, Top/Side/Front views will retain Orthographic or Perspective view (whichever was active at the time of switching to that view).

Smooth View

Length of time the animation takes when changing the view with the numpad (Top/Side/Front/Camera...). Reduce to zero to remove the animation.

Rotation Angle

Rotation step size in degrees, when 4 NumPad, 6 NumPad, 8 NumPad, or 2 NumPad are used to rotate the 3D view.

2D Ansichtspunkte

Minimum Grid Spacing

Die minimale Anzahl an Pixel zwischen Gitterlines in einem (z.B. top orthographischen) Ansichtspunkt.

TimeCode Style

Format des Zeit Codes das angezeigt wird wenn nicht Frames als Format für die Zeit genutzt wird. Das Format benutzt '+' als Seperator für Unter-Sekunden Framenummern, mit linker und rechter Abschneidung des Zeit Codes wenn notwendig.

Manipulator

Lässt die Konfiguration vom 3D Transform Manipulator zu, welcher zum ziehen, rotieren und vergrößern von Objekten benutzt wird (Größe, Handle Size).

Menüs

Beim Mausüberfahren öffnen

Wählen Sie diesen Punkt aus, um das Menü durch die Platzierung des Mauszeigers über dem Eintrag zu bewirken, anstatt auf den Eintrag zu klicken.

Menu Open Delay

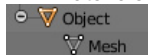
Top Level

Zeitverzögerung in 1/10 Sekunden bevor sich eine Menü öffnet (Beim Mausüberfahren öffnen muss dafür aktiviert sein).

Sub Level

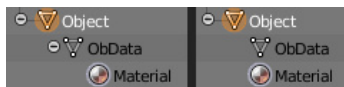
Same as above for sub menus (for example: File » Open Recent).

Link Materials To



To understand this option properly, you need to understand how Blender works with Objects. Almost everything in Blender is organized in a hierarchy of Datablocks. A Datablock can be thought of as containers for certain pieces of information. For example, the Object Datablock contains information about the Object's location while the Object Data (ObData) datablock contains information about the mesh.

A material may be linked in two different ways:



A material linked to ObData (left) and Object (right).

ObData

Any created material will be created as part of the ObData datablock.

Object

Any created material will be created as part of the Object datablock.

[Read more about Blender's Data System »](#)

New objects

Enter Edit Mode

If selected, Edit Mode is automatically activated when you create a new object.

Align To

World

New objects align with world coordinates.

View

New object align with view coordinates.

Undo

Global Undo

Works by keeping a full copy of the file in memory (thus needing more memory).

Step

Number of Undo steps available.

Memory Limit

Maximum memory usage in Mb (0 is unlimited).

[Read more about Undo and Redo options »](#)

Grease Pencil

Grease Pencil permits you to draw in the 3D viewport with a pencil-like tool.

Manhattan Distance

The minimum number of pixels the mouse has to move horizontally or vertically before the movement is recorded.

Euclidian Distance

The minimum distance that mouse has to travel before movement is recorded.

Eraser Radius

The size of the eraser used with the grease pencil.

Smooth Stroke

Smooths the pencil stroke after it's finished.

Playback

Allow Negative Frame

If set, negative framenumbers might be used.

Keyframing

In many situations, animation is controlled by keyframes. The state of a value (i.e. location) is recorded in a keyframe and the animation between two keyframes is interpolated by Blender.

Visual Keying

Use Visual keying automatically for constrained objects.

Only Insert Needed

When enabled, new keyframes will be created only when needed.

Auto Keyframing

Automatic keyframe insertion for Objects and Bones. Auto Keyframe is not enabled by default.
Only Insert Available

Automatic keyframe insertion in available curves.

New F-Curve Defaults Interpolation




This controls how the state between two keyframes is computed. Default interpolation for new keyframes is Bezier which provides smooth acceleration and de-acceleration whereas Linear or Constant is more abrupt.

XYZ to RGB

Color for X, Y or Z animation curves (location, scale or rotation) are the same as the colour for the X, Y and Z axis.

Transform

Release confirm

Dragging LMB  on an object will move it. To confirm this (and other) transforms, a LMB  is necessary by default. When this option is activated, the release of LMB  acts as confirmation of the transform.

Duplicate Data

The 'Duplicate Data' check-boxes define what data is copied with a duplicated Object and what data remains linked. Any boxes that are checked will have their data copied along with the duplication of the Object. Any boxes that are not checked will instead have their data linked from the source Object that was duplicated.

For example, if you have Mesh checked, then a full copy of the mesh data is created with the new Object, and each mesh will behave independently of the duplicate. If you leave the mesh box unchecked then when you change the mesh of one object, the change will be mirrored in the duplicate Object.

The same rules apply to each of the check-boxes in the 'Duplicate Data' list.

Voreinstellungen verwalten

Blender lässt Sie mehrere Eingabeeinstellungen zur Voreinstellung festlegen. Anstatt die Standard Tastenkombinationen zu löschen und Ihre eigenen festlegen zu können, können Sie eine neue Voreinstellungen für Maus und Tastatur festlegen. Optionen zur Maus können auf der linken Handseite des Fensters, und Optionen zur Tastatur auf der rechten Seite im obenstehendem Bild gefunden werden.

Hinzufügen und Entfernen von Voreinstellungen



Bevor Sie irgendetwas in der Standardkonfiguration ändern, klicken Sie auf das "Plus" Symbol das im Bild gezeigt wird um eine neue Voreinstellung hinzuzufügen. Blender wird Sie fragen Ihre Voreinstellung zu benennen, nachdem Sie Ihre Voreinstellung aus der Liste auswählen können. Falls Sie Ihre Voreinstellung entfernen möchten, wählen Sie sie aus der Liste und klicken Sie auf das "Minus" Symbol..

Voreinstellungen auswählen

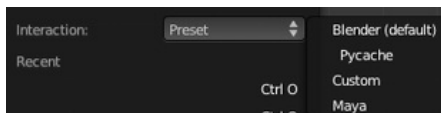
You can change the preset you are using by doing one of the following:

- Selecting the configuration from the Interaction menu of the splash screen at startup or by selecting Help » Splash Screen.
- Selecting the configuration from the User Preferences Input window.

Note

Note that either of the above options will only change the preset for the current file. If you select File » New or File » Open, the default preset will be re-loaded.

Voreinstellung als Standard festlegen



Once you've configured your mouse and keyboard Presets, you can make this the default configuration by:

- Opening the User Preferences Input editor and select your presets from the preset list or,
- Selecting your preset configuration from the splash screen.
- Saving your configuration using the Save As Default option from a User Preferences window or by pressing CtrlU.

Tastenkombinationen Export/Import

In some cases, you may need to save your configuration in an external file (e.g. if you need to install a new system or share your keymap configuration with the community). Simply LMB Export Key Configuration on the Input tab header and a file browser will open so that you can choose where to store the configuration. The Import Key Configuration button installs a keymap configuration that is on your computer but not in Blender.

Maus

Emulate 3 Button Mouse

It is possible to use Blender without a 3 button mouse (such as a two-button mouse, Apple single-button Mouse, or laptop). This functionality can be emulated with key/mousebutton combos. This option is only available if Select With is set to Right.

[Read more about emulating a 3 button mouse »](#)

Continuous Grab

Allows moving the mouse outside of the view (for translation, rotation, scale for example).

Drag Threshold

The number of pixels that a User Interface element has to be moved before it is recognized by Blender.

Select with

You can choose which button is used for selection (the other one is used to place the 3D cursor).

Double Click

The time for a double click (in ms).

Note


If you're using a graphic tablet instead of mouse, and pressure doesn't work properly, try to place the mouse pointer to Blender window and then unplug/replug your graphic tablet. This might help.

Numpad Emulation

The Numpad keys are used quite often in Blender and are not the same keys as the regular number keys. If you have a keyboard without a Numpad (e.g. on a laptop), you can tell Blender to treat the standard number keys as Numpad keys. Just check Emulate Numpad.

Manipulation der Ansicht



Orbit Style

Select how Blender works when you rotate the 3D view (by default MMB ). Two styles are available. If you come from Maya or Cinema 4D, you will prefer Turntable.


Zoom Style

Choose your preferred style of zooming in and out with Ctrl MMB .

Scale

Scale zooming depends on where you first click in the view. To zoom out, hold Ctrl MMB  while dragging from the edge of the screen towards the center. To zoom in, hold Ctrl MMB  while dragging from the center of the screen towards the edge.

Continue

The Continue zooming option allows you to control the speed (and not the value) of zooming by moving away from the initial click-point with Ctrl MMB . Moving up from the initial click-point or to the right will zoom out, moving down or to the left will zoom in. The further away you move, the faster the zoom movement will be. The directions can be altered by the Vertical and Horizontal radio buttons and the Invert Zoom Direction option.

Dolly

Dolly zooming works similarly to Continue zooming except that zoom speed is constant.

Vertical

Moving up zooms out and moving down zooms in.

Horizontal

Moving left zooms in and moving right zooms out.

Invert Zoom Direction

Inverts the Zoom direction for Dolly and Continue zooming.

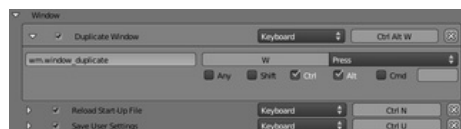
Invert Wheel Zoom Direction

Inverts the direction of the mouse wheel zoom.

NDOF device

Set the sensitivity of a 3D mouse.

Tastenkombinationen Editor



Der Tastenkombinationen Editor lässt Sie die Standard Tastenkombinationen ändern. Sie können die Tastenkombinationen für jedes Fenster ändern.

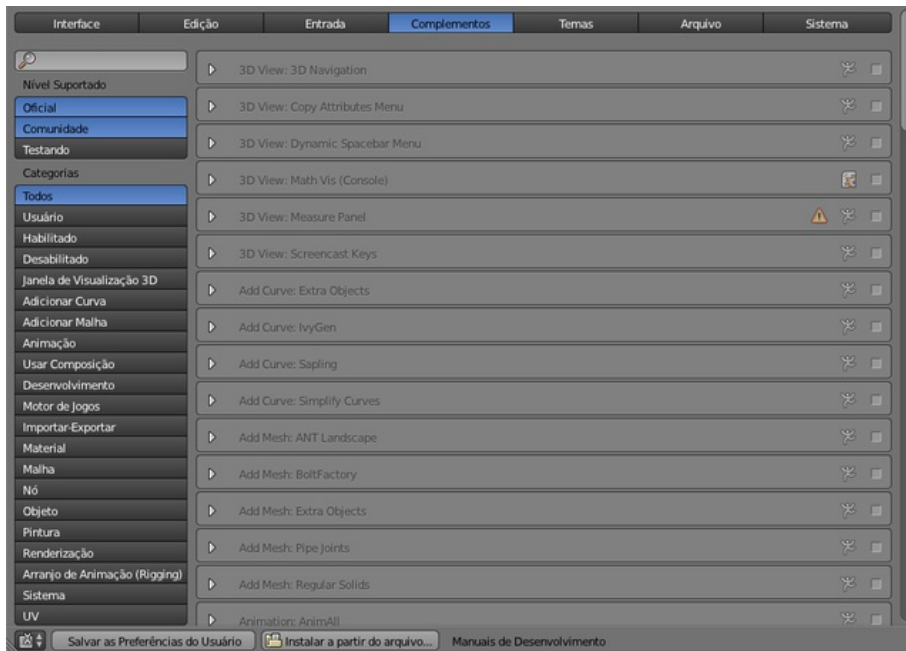
1. Select the keymap you want to change and click on the white arrows to open up the keymap tree.
2. Select which Input will control the function
 - Keyboard: Only hotkey or combo hotkey (E or ⇧ ShiftE).
 - Mouse: Left/middle/right click. Can be combined with Alt, ⇧ Shift, Ctrl, ⌘ Cmd.
 - Tweak: Click and drag. Can also be combined with the 4 previous keys.
 - Text input: Use this function by entering a text
 - Timer: Used to control actions based on a time period. e.g. By default, Animation Step uses Timer 0, Smooth view uses Timer 1.
3. Change hotkeys as you want. Just click on the shortcut input and enter the new shortcut.

If you want to restore the default settings for a keymap, just click on the Restore button at the top right of this keymap.

[Oberfläche](#) • [Editieren](#) • [Eingabe](#) • **Addons** • [Themen](#) • [Datei](#) • [System](#)

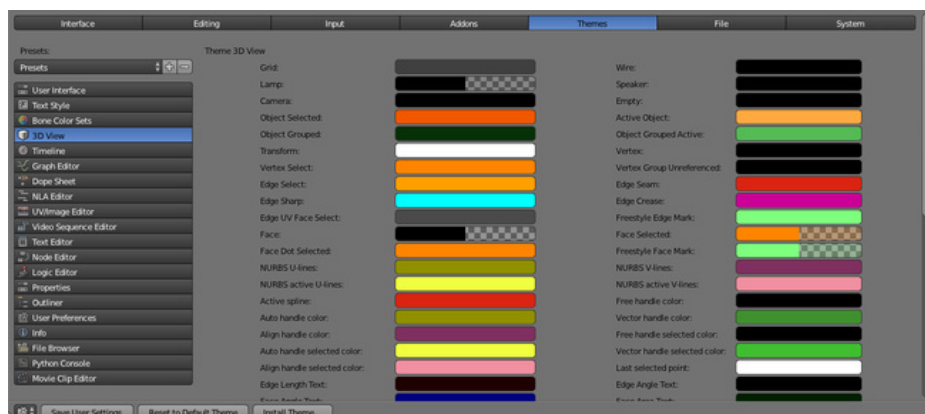
The Add-Ons tab lets you manage secondary options which are not enabled in Blender by default. New features may be added with *Install Add-Ons*. There will be a growing number of such Add-Ons, generated by the Blender-community so look out for that one feature you were missing (or maybe simply create it yourself).

See the [Add-Ons Page](#) for more on using Add-Ons.



Customizing themes

As has been said previously, Blender is very customizable—some of the settings affect interface appearance and colors. These are set under the Themes tab.



The colors for each editor can be set separately—simply select the editor you wish to change in the multi-choice list at the left, and adjust colors as required. Notice that changes appear in real-time on your screen. In addition, details such as the dot size in the 3D View or the Graph Editor can also be changed.

Customizing themes with Python

If you would like to control the theme from a script, here is an example script with the parameters listed individually:

```
# A simple way to make global theme color changes.
```

```
import bpy
```

```
#Shared elements
```

```
back = (0.2667,0.2667,0.2667)
viewbg = (0.2667,0.3,0.34)
back_alt = (0.1647,0.1647,0.1647)
button = (0.2667,0.2667,0.2667)
grid = (0.5,0.5,0.5)
header = (0.2667,0.2667,0.2667)
panel = (0.2667,0.2667,0.2667,0.75)
text = (0.8,0.8,0.8)
text_hi = (1,1,1)
wire = (1,1,1)
lamp = (1,1,1,1)
```

```
#UI shared
```

```
inner = (0.2667,0.2667,0.2667,1)
inner_sel = (0.2667,0.2667,0.2667,1)
item = (0.39,0.39,0.39,1)
outline = (0.1,0.1,0.1)
```

```
theme = bpy.context.user_preferences.themes['Default']
```

```
theme.view_3d.act_spline
theme.view_3d.back = viewbg
theme.view_3d.bone_pose
theme.view_3d.bone_solid
theme.view_3d.button= button
theme.view_3d.button_text = text
theme.view_3d.button_text_hi = text_hi
theme.view_3d.button_title = text
theme.view_3d.edge_crease
theme.view_3d.edge_facesel
theme.view_3d.edge_seam
theme.view_3d.edge_select
theme.view_3d.edge_sharp
theme.view_3d.editmesh_active
theme.view_3d.face
theme.view_3d.face_dot
theme.view_3d.face_select
theme.view_3d.facedot_size
theme.view_3d.frame_current
theme.view_3d.grid = grid
theme.view_3d.handle_align
theme.view_3d.handle_auto
theme.view_3d.handle_free
theme.view_3d.handle_sel_align
theme.view_3d.handle_sel_auto
theme.view_3d.handle_sel_free
theme.view_3d.handle_sel_vect
theme.view_3d.handle_vect
theme.view_3d.header = header
theme.view_3d.header_text = text
theme.view_3d.header_text_hi = text_hi
theme.view_3d.lamp = lamp
```

```
theme.view_3d.lastsel_point
theme.view_3d.normal
theme.view_3d.nurb_sel_uline
theme.view_3d.nurb_sel_vline
theme.view_3d.nurb_uline
theme.view_3d.nurb_vline
theme.view_3d.object_active
theme.view_3d.object_grouped
theme.view_3d.object_grouped_active
theme.view_3d.object_selected
theme.view_3d.panel = panel
theme.view_3d.text = text
theme.view_3d.text_hi = text_hi
theme.view_3d.title = text
theme.view_3d.transform
theme.view_3d.vertex
theme.view_3d.vertex_normal
theme.view_3d.vertex_select
theme.view_3d.vertex_size
theme.view_3d.wire = wire

theme.console.back= back_alt
theme.console.button = button
theme.console.button_text = text
theme.console.button_text_hi = text_hi
theme.console.button_title = text
theme.console.cursor
theme.console.header = header
theme.console.header_text = text
theme.console.header_text_hi = text_hi
theme.console.line_error
theme.console.line_info
theme.console.line_input
theme.console.line_output
theme.console.text = text
theme.console.text_hi = text_hi
theme.console.title = text

theme.dopesheet_editor.active_channels_group
theme.dopesheet_editor.back= back
theme.dopesheet_editor.button = button
theme.dopesheet_editor.button_text = text
theme.dopesheet_editor.button_text_hi = text_hi
theme.dopesheet_editor.button_title = text
theme.dopesheet_editor.channel_group
theme.dopesheet_editor.channels
theme.dopesheet_editor.channels_selected
theme.dopesheet_editor.dopesheet_channel
theme.dopesheet_editor.dopesheet_subchannel
theme.dopesheet_editor.frame_current
theme.dopesheet_editor.grid = grid
theme.dopesheet_editor.header = header
theme.dopesheet_editor.header_text = text
theme.dopesheet_editor.header_text_hi = text_hi
theme.dopesheet_editor.list = button
theme.dopesheet_editor.list_text = text
theme.dopesheet_editor.list_text_hi = text
theme.dopesheet_editor.list_title = text
theme.dopesheet_editor.long_key
theme.dopesheet_editor.long_key_selected
theme.dopesheet_editor.text = text
theme.dopesheet_editor.text_hi = text_hi
theme.dopesheet_editor.title = text
theme.dopesheet_editor.value_sliders
theme.dopesheet_editor.view_sliders

theme.file_browser.active_file
theme.file_browser.active_file_text
theme.file_browser.back= back
theme.file_browser.button = button
theme.file_browser.button_text = text
theme.file_browser.button_text_hi = text_hi
theme.file_browser.button_title = text
theme.file_browser.header = header
theme.file_browser.header_text = text
theme.file_browser.header_text_hi = text_hi
theme.file_browser.list = button
theme.file_browser.list_text = text
theme.file_browser.list_text_hi = text
theme.file_browser.list_title = text
theme.file_browser.scroll_handle = button
theme.file_browser.scrollbar = button
theme.file_browser.selected_file
theme.file_browser.text = text
theme.file_browser.text_hi = text_hi
theme.file_browser.tiles = back
theme.file_browser.title = text

theme.graph_editor.active_channels_group
```



```
theme.graph_editor.back= back_alt
theme.graph_editor.button = button
theme.graph_editor.button_text = text
theme.graph_editor.button_text_hi = text_hi
theme.graph_editor.button_title = text
theme.graph_editor.channel_group
theme.graph_editor.channels_region
theme.graph_editor.dopesheet_channel
theme.graph_editor.dopesheet_subchannel
theme.graph_editor.frame_current
theme.graph_editor.grid = grid
theme.graph_editor.handle_align
theme.graph_editor.handle_auto
theme.graph_editor.handle_free
theme.graph_editor.handle_sel_align
theme.graph_editor.handle_sel_auto
theme.graph_editor.handle_sel_free
theme.graph_editor.handle_sel_vect
theme.graph_editor.handle_vect
theme.graph_editor.handle_vertex
theme.graph_editor.handle_vertex_select
theme.graph_editor.handle_vertex_size
theme.graph_editor.header = header
theme.graph_editor.header_text = text
theme.graph_editor.header_text_hi = text_hi
theme.graph_editor.lastsel_point
theme.graph_editor.list = button
theme.graph_editor.list_text = text
theme.graph_editor.list_text_hi = text
theme.graph_editor.list_title = text
theme.graph_editor.panel = back
theme.graph_editor.text = text
theme.graph_editor.text_hi = text_hi
theme.graph_editor.title = text
theme.graph_editor.vertex
theme.graph_editor.vertex_select
theme.graph_editor.vertex_size
theme.graph_editor.window_sliders
```

```
theme.image_editor.back= back_alt
theme.image_editor.button = button
theme.image_editor.button_text = text
theme.image_editor.button_text_hi = text_hi
theme.image_editor.button_title = text
theme.image_editor.editmesh_active
theme.image_editor.face
theme.image_editor.face_dot
theme.image_editor.face_select
theme.image_editor.facedot_size
theme.image_editor.header = header
theme.image_editor.header_text = text
theme.image_editor.header_text_hi = text_hi
theme.image_editor.scope_back
theme.image_editor.text = text
theme.image_editor.text_hi = text_hi
theme.image_editor.title = text
theme.image_editor.vertex
theme.image_editor.vertex_select
theme.image_editor.vertex_size
```

```
theme.info.back= back
theme.info.button = button
theme.info.button_text = text
theme.info.button_text_hi = text_hi
theme.info.button_title = text
theme.info.header = header
theme.info.header_text = text
theme.info.header_text_hi = text_hi
theme.info.text = text
theme.info.text_hi = text_hi
theme.info.title = text
```

```
theme.logic_editor.back= back
theme.logic_editor.button = button
theme.logic_editor.button_text = text
theme.logic_editor.button_text_hi = text_hi
theme.logic_editor.button_title = text
theme.logic_editor.header = header
theme.logic_editor.header_text = text
theme.logic_editor.header_text_hi = text_hi
theme.logic_editor.panel = back
theme.logic_editor.text = text
theme.logic_editor.text_hi = text_hi
theme.logic_editor.title = text
```

```
theme.nla_editor.back= back
theme.nla_editor.bars
```

```
theme.nla_editor.bars_selected
theme.nla_editor.button = button
theme.nla_editor.button_text = text
theme.nla_editor.button_text_hi = text_hi
theme.nla_editor.button_title = text
theme.nla_editor.frame_current
theme.nla_editor.grid = grid
theme.nla_editor.header = header
theme.nla_editor.header_text = text
theme.nla_editor.header_text_hi = text_hi
theme.nla_editor.list = button
theme.nla_editor.list_text = text
theme.nla_editor.list_text_hi = text
theme.nla_editor.list_title = text
theme.nla_editor.strips
theme.nla_editor.strips_selected
theme.nla_editor.text = text
theme.nla_editor.text_hi = text_hi
theme.nla_editor.title = text
theme.nla_editor.view_sliders

theme.node_editor.back= back_alt
theme.node_editor.button = button
theme.node_editor.button_text = text
theme.node_editor.button_text_hi = text_hi
theme.node_editor.button_title = text
theme.node_editor.converter_node
theme.node_editor.group_node
theme.node_editor.header = header
theme.node_editor.header_text = text
theme.node_editor.header_text_hi = text_hi
theme.node_editor.in_out_node
theme.node_editor.list = button
theme.node_editor.list_text = text
theme.node_editor.list_text_hi = text
theme.node_editor.list_title = text
theme.node_editor.node_backdrop
theme.node_editor.operator_node
theme.node_editor.selected_text
theme.node_editor.text = text
theme.node_editor.text_hi = text_hi
theme.node_editor.title = text
theme.node_editor.wire
theme.node_editor.wire_select

theme.outliner.back= back
theme.outliner.button = button
theme.outliner.button_text = text
theme.outliner.button_text_hi = text_hi
theme.outliner.button_title = text
theme.outliner.header = header
theme.outliner.header_text = text
theme.outliner.header_text_hi = text_hi
theme.outliner.text = text
theme.outliner.text_hi = text_hi
theme.outliner.title = text

theme.properties.back= back
theme.properties.button = button
theme.properties.button_text = text
theme.properties.button_text_hi = text_hi
theme.properties.button_title = text
theme.properties.header = header
theme.properties.header_text = text
theme.properties.header_text_hi = text_hi
theme.properties.panel = back
theme.properties.text = text
theme.properties.text_hi = text_hi
theme.properties.title = text

theme.sequence_editor.audio_strip
theme.sequence_editor.back= back
theme.sequence_editor.button = button
theme.sequence_editor.button_text = text
theme.sequence_editor.button_text_hi = text_hi
theme.sequence_editor.button_title = text
theme.sequence_editor.draw_action
theme.sequence_editor.effect_strip
theme.sequence_editor.frame_current
theme.sequence_editor.grid = grid
theme.sequence_editor.header = header
theme.sequence_editor.header_text = text
theme.sequence_editor.header_text_hi = text_hi
theme.sequence_editor.image_strip
theme.sequence_editor.keyframe
theme.sequence_editor.meta_strip
theme.sequence_editor.movie_strip
theme.sequence_editor.plugin_strip
```

```

theme.sequence_editor.scene_strip
theme.sequence_editor.text = text
theme.sequence_editor.text_hi = text_hi
theme.sequence_editor.title = text
theme.sequence_editor.transition_strip
theme.sequence_editor.window_sliders

```

```

theme.text_editor.back= back_alt
theme.text_editor.button = button
theme.text_editor.button_text = text
theme.text_editor.button_text_hi = text_hi
theme.text_editor.button_title = text
theme.text_editor.cursor
theme.text_editor.header = header
theme.text_editor.header_text = text
theme.text_editor.header_text_hi = text_hi
theme.text_editor.line_numbers_background
theme.text_editor.scroll_bar = back
theme.text_editor.selected_text
theme.text_editor.syntax_builtin
theme.text_editor.syntax_comment
theme.text_editor.syntax_numbers
theme.text_editor.syntax_special
theme.text_editor.syntax_string
theme.text_editor.text = text
theme.text_editor.text_hi = text_hi
theme.text_editor.title = text

```

```

theme.timeline.back= back_alt
theme.timeline.button = button
theme.timeline.button_text = text
theme.timeline.button_text_hi = text_hi
theme.timeline.button_title = text
theme.timeline.frame_current
theme.timeline.grid = grid
theme.timeline.header = header
theme.timeline.header_text = text
theme.timeline.header_text_hi = text_hi
theme.timeline.text = text
theme.timeline.text_hi = text_hi
theme.timeline.title = text

```

```

theme.user_preferences.back= back
theme.user_preferences.button = button
theme.user_preferences.button_text = text
theme.user_preferences.button_text_hi = text_hi
theme.user_preferences.button_title = text
theme.user_preferences.header = header
theme.user_preferences.header_text = text
theme.user_preferences.header_text_hi = text_hi
theme.user_preferences.text = text
theme.user_preferences.text_hi = text_hi
theme.user_preferences.title = text

```

```

theme.user_interface.icon_file

```

```

theme.user_interface.wcol_box.inner = inner
theme.user_interface.wcol_box.inner_sel
theme.user_interface.wcol_box.item = item
theme.user_interface.wcol_box.outline = outline
theme.user_interface.wcol_box.shadedown
theme.user_interface.wcol_box.shadetop
theme.user_interface.wcol_box.show_shaded
theme.user_interface.wcol_box.text = text
theme.user_interface.wcol_box.text_sel = text_hi

```

```

theme.user_interface.wcol_list_item.inner = inner
theme.user_interface.wcol_list_item.inner_sel
theme.user_interface.wcol_list_item.item = item
theme.user_interface.wcol_list_item.outline = outline
theme.user_interface.wcol_list_item.shadedown
theme.user_interface.wcol_list_item.shadetop
theme.user_interface.wcol_list_item.show_shaded
theme.user_interface.wcol_list_item.text = text
theme.user_interface.wcol_list_item.text_sel = text_hi

```

```

theme.user_interface.wcol_menu.inner = inner
theme.user_interface.wcol_menu.inner_sel
theme.user_interface.wcol_menu.item = item
theme.user_interface.wcol_menu.outline = outline
theme.user_interface.wcol_menu.shadedown
theme.user_interface.wcol_menu.shadetop
theme.user_interface.wcol_menu.show_shaded
theme.user_interface.wcol_menu.text = text
theme.user_interface.wcol_menu.text_sel = text_hi

```

```

theme.user_interface.wcol_menu_back.inner = inner

```

```
theme.user_interface.wcol_menu_back.inner_sel
theme.user_interface.wcol_menu_back.item = item
theme.user_interface.wcol_menu_back.outline = outline
theme.user_interface.wcol_menu_back.shadedown
theme.user_interface.wcol_menu_back.shadetop
theme.user_interface.wcol_menu_back.show_shaded
theme.user_interface.wcol_menu_back.text = text
theme.user_interface.wcol_menu_back.text_sel = text_hi
```

```
theme.user_interface.wcol_menu_item.inner
theme.user_interface.wcol_menu_item.inner_sel
theme.user_interface.wcol_menu_item.item = item
theme.user_interface.wcol_menu_item.outline = outline
theme.user_interface.wcol_menu_item.shadedown
theme.user_interface.wcol_menu_item.shadetop
theme.user_interface.wcol_menu_item.show_shaded
theme.user_interface.wcol_menu_item.text = text
theme.user_interface.wcol_menu_item.text_sel = text_hi
```

```
theme.user_interface.wcol_num.inner = inner
theme.user_interface.wcol_num.inner_sel
theme.user_interface.wcol_num.item = item
theme.user_interface.wcol_num.outline = outline
theme.user_interface.wcol_num.shadedown
theme.user_interface.wcol_num.shadetop
theme.user_interface.wcol_num.show_shaded
theme.user_interface.wcol_num.text = text
theme.user_interface.wcol_num.text_sel = text_hi
```

```
theme.user_interface.wcol_numslider.inner = inner
theme.user_interface.wcol_numslider.inner_sel
theme.user_interface.wcol_numslider.item = item
theme.user_interface.wcol_numslider.outline = outline
theme.user_interface.wcol_numslider.shadedown
theme.user_interface.wcol_numslider.shadetop
theme.user_interface.wcol_numslider.show_shaded
theme.user_interface.wcol_numslider.text = text
theme.user_interface.wcol_numslider.text_sel = text_hi
```

```
theme.user_interface.wcol_option.inner = inner
theme.user_interface.wcol_option.inner_sel
theme.user_interface.wcol_option.item = item
theme.user_interface.wcol_option.outline = outline
theme.user_interface.wcol_option.shadedown
theme.user_interface.wcol_option.shadetop
theme.user_interface.wcol_option.show_shaded
theme.user_interface.wcol_option.text = text
theme.user_interface.wcol_option.text_sel = text_hi
```

```
theme.user_interface.wcol_progress.inner = inner
theme.user_interface.wcol_progress.inner_sel
theme.user_interface.wcol_progress.item = item
theme.user_interface.wcol_progress.outline = outline
theme.user_interface.wcol_progress.shadedown
theme.user_interface.wcol_progress.shadetop
theme.user_interface.wcol_progress.show_shaded
theme.user_interface.wcol_progress.text = text
theme.user_interface.wcol_progress.text_sel = text_hi
```

```
theme.user_interface.wcol_pulldown.inner = inner
theme.user_interface.wcol_pulldown.inner_sel
theme.user_interface.wcol_pulldown.item = item
theme.user_interface.wcol_pulldown.outline = outline
theme.user_interface.wcol_pulldown.shadedown
theme.user_interface.wcol_pulldown.shadetop
theme.user_interface.wcol_pulldown.show_shaded
theme.user_interface.wcol_pulldown.text = text
theme.user_interface.wcol_pulldown.text_sel = text_hi
```

```
theme.user_interface.wcol_radio.inner = inner
theme.user_interface.wcol_radio.inner_sel
theme.user_interface.wcol_radio.item = item
theme.user_interface.wcol_radio.outline = outline
theme.user_interface.wcol_radio.shadedown
theme.user_interface.wcol_radio.shadetop
theme.user_interface.wcol_radio.show_shaded
theme.user_interface.wcol_radio.text = text
theme.user_interface.wcol_radio.text_sel = text_hi
```

```
theme.user_interface.wcol_regular.inner = inner
theme.user_interface.wcol_regular.inner_sel
theme.user_interface.wcol_regular.item = item
theme.user_interface.wcol_regular.outline = outline
theme.user_interface.wcol_regular.shadedown
theme.user_interface.wcol_regular.shadetop
theme.user_interface.wcol_regular.show_shaded
theme.user_interface.wcol_regular.text = text
theme.user_interface.wcol_regular.text_sel = text_hi
```

```
theme.user_interface.wcol_scroll.inner = inner
theme.user_interface.wcol_scroll.inner_sel
theme.user_interface.wcol_scroll.item = item = inner
```

```
theme.user_interface.wcol_scroll.outline = outline
theme.user_interface.wcol_scroll.shadedown
theme.user_interface.wcol_scroll.shadetop
theme.user_interface.wcol_scroll.show_shaded
theme.user_interface.wcol_scroll.text = text
theme.user_interface.wcol_scroll.text_sel = text_hi
```

```
theme.user_interface.wcol_state.inner_anim
theme.user_interface.wcol_state.inner_anim_sel
theme.user_interface.wcol_state.inner_driven
theme.user_interface.wcol_state.inner_driven_sel
theme.user_interface.wcol_state.inner_key
theme.user_interface.wcol_state.inner_key_sel
```

```
theme.user_interface.wcol_text.inner = inner
theme.user_interface.wcol_text.inner_sel
theme.user_interface.wcol_text.item = item
theme.user_interface.wcol_text.outline = outline
theme.user_interface.wcol_text.shadedown
theme.user_interface.wcol_text.shadetop
theme.user_interface.wcol_text.show_shaded
theme.user_interface.wcol_text.text = text
theme.user_interface.wcol_text.text_sel = text_hi
```

```
theme.user_interface.wcol_toggle.inner = inner
theme.user_interface.wcol_toggle.inner_sel
theme.user_interface.wcol_toggle.item = item
theme.user_interface.wcol_toggle.outline = outline
theme.user_interface.wcol_toggle.shadedown
theme.user_interface.wcol_toggle.shadetop
theme.user_interface.wcol_toggle.show_shaded
theme.user_interface.wcol_toggle.text = text
theme.user_interface.wcol_toggle.text_sel = text_hi
```

```
theme.user_interface.wcol_tool.inner = inner
theme.user_interface.wcol_tool.inner_sel
theme.user_interface.wcol_tool.item = item
theme.user_interface.wcol_tool.outline = outline
theme.user_interface.wcol_tool.shadedown
theme.user_interface.wcol_tool.shadetop
theme.user_interface.wcol_tool.show_shaded
theme.user_interface.wcol_tool.text = text
theme.user_interface.wcol_tool.text_sel = text_hi
```

File Preferences

Das Bild zeigt die Datei-Eigenschaften die weiter unten erklärt werden.



Datei Pfade

Wenn Sie an einem importiertem Projekt arbeiten, ist es schlaun dieses zu konfigurieren. Setzen Sie die Standard Pfade für die verschiedenen Dateitypen die Sie nutzen werden.

Hier ist ein Beispiel einer Konfiguration

Schriften	//fonts/
Texturen	//textures/
Textur Plugins	//plugins/texture/
Sequence Plugins	//plugins/sequence/
Render Ausgabe	//renders/
Skripte	//scripts/
Klänge	//sounds/
Temp	//tmp/

Beachten Sie dass Blender Ihre Projekt Struktur nicht automatisch anlegt. Sie müssen alle Verzeichnisse manuell in Ihrer Dateiverwaltung anlegen.

Speichern & Laden

Relative Pfade

Standardmäßig, nutzen externe Dateien einen relativen Pfad. Dies funktioniert nur wenn eine Blender Datei gespeichert ist.

Datei komprimieren

Komprimieren der .blend Datei wenn diese gespeichert wird.

Benutzeroberfläche laden

Standard ist die Benutzeroberfläche zu laden (die [Bildschirme](#)) der gespeicherte Datei. Dies kann individuell geändert werden wenn eine Datei aus dem Blender-Datei öffnen Panel des Fensters der Dateiverwaltung.



Datei Erweiterungs Filter

Dateierweiterungen filtern

Durch die Aktivierung, werden Datei Dialog Fenster nur die passenden Dateien anzeigen (z.B. .blend Dateien wenn eine komplette Blender Eigenschaft geladen wird). Die Auswahl an Dateitypen kann im Datei Dialog Fenster verändert werden.

Punkt-Dateien/Datenblöcke verstecken

Dateien verstecken die mit "." in Dateibrowsern beginnen (in Linux und Apple Systemen, sind "." Dateien versteckt).

Hide Recent Locations

Versteckt die kürzlich verwendet Panel des Fensters der Dateiverwaltung welches Verzeichnisse anzeigt auf die kürzlich zugegriffen wurde.

Vorschaubilder anzeigen

Zeigt Vorschaubilder von Bildern und Filmen an wenn die Dateiverwaltung genutzt wird.

Automatische Sicherung

Save Versions

Anzahl an Versionen die für die selbe Datei erstellt werden (zur Datensicherung).

Kürzlich geöffnete Dateien

Anzahl an Dateien die in Datei » Kürzlich geöffnet angezeigt werden.

Vorschaubilder speichern

Vorschaubilder von Bildern und Materialien in dem Fenster der Dateiverwaltung werden nach Bedarf generiert. Zum Speichern von Vorschaubildern in Ihre `.blend` Datei, aktivieren Sie diese Operation (auf Kosten der Vergrößerung Ihrer `.blend` Datei).

Temporäre Dateien automatisch speichern

Automatisches Speichern aktivieren (erstellt eine temporäre Datei).

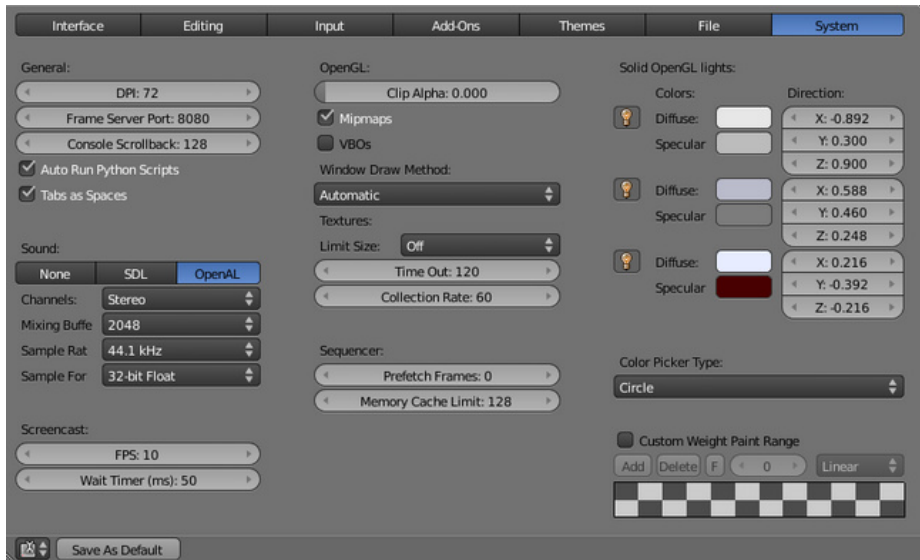
Intervall

Zeit die zwischen den automatischen Speichervorgängen gewartet wird.

[Mehr über Operationen zur Automatischen Sicherung »](#)

System preferences

The picture shows the System tab in the User Preferences window Blender. The several options are explained below.



General

DPI

This value is the screen resolution and thus controls the size of the interface fonts. To change the size of parts of the interface, you might prefer pressing Ctrl and dragging MMB left and right over a panel to resize its contents.

Frame Server Port

Frameserver port for frameserver rendering. Used when working with distributed rendering.

Console Scrollback

The number of lines, buffered in memory of the console window.

Auto Run Python Scripts

Allows .blend files to run their scripts automatically (might be unsafe with files from unknown sources).

Tabs as Spaces

Tabs in text files are converted into spaces while typing or when a text file is loaded.

Sound

Sound

Set the audio output device.

Channels

Set the audio channel count.

Mixing Buffer

Set the number of samples used by the audio mixing buffer.

Sample Rate

Set the audio sample rate.

Sample Format

Set the audio sample format.

Screencast

FPS

Framerate for the screencast to be played back.

Wait Timer

Time in milliseconds between each frame recorded for screencasts.

Open GL

Clip Alpha

Clip alpha below this threshold in the 3D viewport.

Mipmaps

Scale textures for 3D view using mipmap filtering. This increases display quality, but uses more memory.

Anisotropic Filtering

Set the level of anisotropic filtering.

VBOs

Use Vertex Buffer Objects, or vertex arrays if unsupported, for viewport rendering.

Window Draw Method

Automatic

Automatically set based on graphics card and driver.

Triple Buffer

Use a third buffer for minimal redraws at the cost of more memory.

Overlap

Redraw all overlapping regions. Minimal memory usage, but more redraws.

Overlap Flip

Redraw all overlapping regions. Minimal memory usage, but more redraws (for graphics drivers that do flipping).

Full

Do a full redraw each time. Only use for reference, or when all else fails.

Text Draw Options

Enable interface text anti-aliasing.

Limit Size

Limit the maximum resolution for pictures used in textured display to save memory.

Time Out

Time since last access of a GL texture in seconds, after which it is freed. Set to 0 to keep textures allocated.

Collection Rate

Number of seconds between each run of the GL texture garbage collector.

Sequencer

Prefetch Frames

Number of frames to render ahead during playback.

Memory Cache Limit


Limit of the sequencer's memory cache (megabytes).

Solid OpenGL lights

When the display in a 3D Window is switched to Solid view, there are three virtual light sources (you won't see them in renders) used to illuminate the scene. Direction, diffuse and specular lighting of these lights can be changed here.

Miscellaneous

Color Picker Type

Choose which type of color dialog you prefer - it will show when clicking LMB  on any color field.

Custom Weight Paint Range

Mesh skin weighting is used to control how much a bone deforms the mesh of a character. To visualize these weights, Blender uses a color ramp (from blue to green to yellow to red). Here you can create your own. With the Controls beneath the options you can add, delete and change the color of Color stops in this ramp.

Unsere erste Animation in 30 + 30 Minuten

Teil I: Ein statischer Lebkuchenmann

In diesem Kapitel werden wir einen kleinen Lebkuchenmann erstellen und animieren. Jeder Schritt wird detailliert beschrieben, aber es wird vorausgesetzt, dass der Inhalt des [Kapitels über die Benutzeroberfläche](#) bereits bekannt ist.

Im ersten Teil dieses Tutorials werden wir zunächst einen *statischen* (noch nicht beweglichen) Pfefferkuchenmann erstellen. Im zweiten Teil bringen wir ihm dann das Laufen bei.

Anmerkung

Eine andere detailliertere Einleitung zu Blender mit Fokus auf Charakter-Animation ist übrigens das [Blender Summer of Documentation Introduction to Character Animation \(engl.\)](#)-Tutorial (ursprünglich für Blender 2.4x, aber immer noch hilfreich).

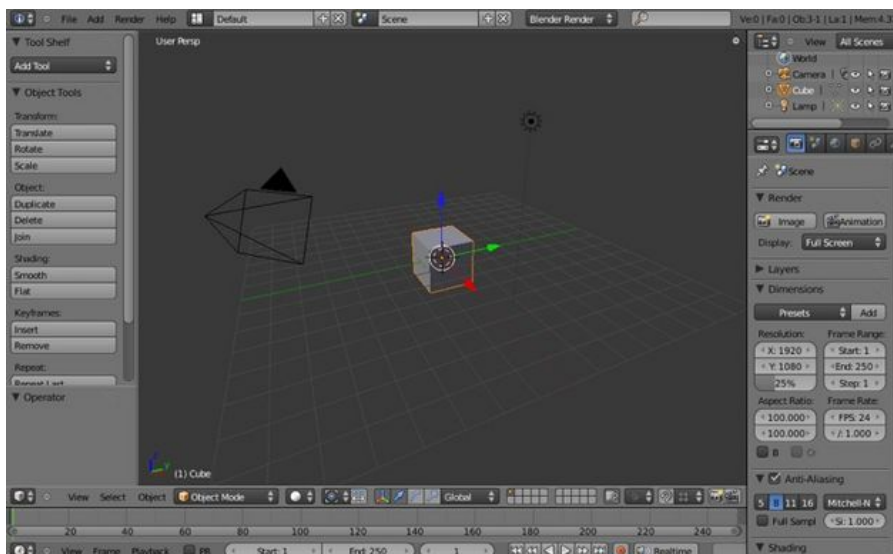


Ähnlich wie "Gus das Lebkuchenmännchen" (nämlich dieses hier) setzt es keine Kenntnisse voraus. Es stellt den Prozess bis zu einem laufenden, sprechenden Charakter von Grund auf dar und behandelt auch schon viele fortgeschrittene Fähigkeiten von Blender, die hier noch außen vor gelassen werden.

Das BSoD Intro kann man auch als [PDF \(3.75 MB\)](#) herunterladen, um es lokal zu lesen.

Aufwärmübungen

Wir starten Blender. Auf dem Bildschirm sollte die Standardszene zu sehen sein: Eine Kamera, ein Licht und ein Würfel. Der Würfel sollte schon ausgewählt sein, was durch seine orangefarbige Umrandung angezeigt wird. (Bild: *Blender nach dem Start*)



Blender nach dem Start

Wir werden die Arbeitsfläche durch das Platzen von Objekten auf verschiedenen Ebenen, auf denen wir sie verstecken können, übersichtlich halten. Man kann diese Objekte leicht wieder in die aktuelle Szene einblenden, sobald wir sie wieder benötigen.

Tastaturkürzel

Blender wird größtenteils über eine Vielzahl von Tastaturkürzeln gesteuert. Die meisten von ihnen funktionieren nur, wenn sich der Mauszeiger über dem entsprechenden Teil des Fensters befindet. Also nicht verzweifeln, falls M (wird etwas weiter unten vorgestellt) nicht das tut, was es tun sollte: Man bewege einfach die Maus in das 3D-Ansichtsfenster.

Nun also ein paar Worte zu Layern (Ebenen):

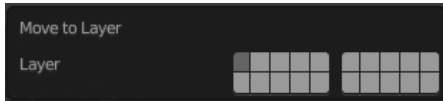


Sichtbarkeit der Layer



Blender stellt zwanzig Layer zur Verfügung, um beim Organisieren zu helfen. Welche Layer im Moment aktiv sind, ist anhand der zwanzig kleinen Schalter in der Header-Leiste des 3D-Ansichtsfensters zu erkennen (Bild: *Sichtbarkeit der Layer*). Den gerade sichtbaren Layer kann man mit LMB auswählen um mehrere Layer gleichzeitig sichtbar zu machen nimmt man \odot Shift LMB zur

Auswahl . Dabei wird der zuletzt aktivierte Layer zum aktiven Layer. Auf diesem Layer werden alle neu erzeugten Objekte gespeichert.

Bevor wir wirklich anfangen, räumen wir erst einmal ein wenig auf.



Layer-Verschiebedialog

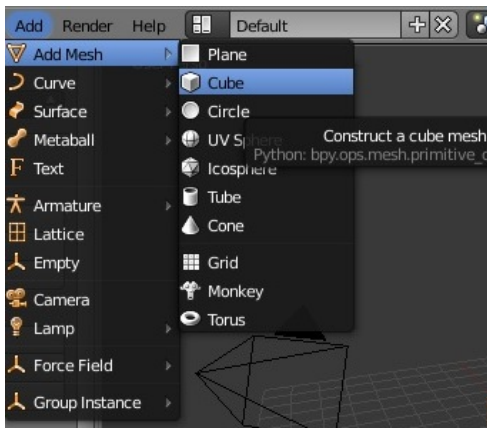
Wir wählen zuerst die Kamera mit RMB  und die Lampe mit ⇧ Shift RMB  aus und drücken danach M. Ein kleiner Dialog (Bild: *Layer-Verschiebedialog*) wird unter der Maus erscheinen. Damit die ausgewählten Objekte auf einen anderen Layer gespeichert werden, wählen wir den Schalter ganz rechts in der oberen Reihe und schon werden Kamera und Lampe auf Layer 10 verschoben.

Nun sollte nur Layer 1 aktiviert sein, damit wir die Lampe oder die Kamera nicht aus Versehen löschen. Alles auf dem Layer mit A auswählen und mit X » löschen » Lösche die ausgewählten Objekte. Jetzt haben wir den nötigen Platz, den wir für das Modelling benötigen werden (und Gus hat auch viel Platz).

Body-Building

Keine Angst, Gus bleibt rank und schlank. Die Überschrift ist nur eine kleine Erinnerung an wunderschöne Übersetzungen. ;)

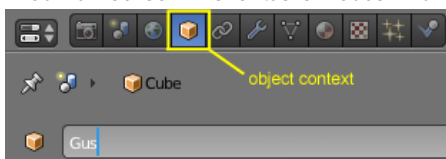
Wir wechseln in die Vorderansicht mit 1 NumPad (Num lock muss aktiviert sein!) und in die orthographische Ansicht mit 5 NumPad, sofern die perspektivische Ansicht ausgewählt sein sollte (angezeigt in der oberen linken Ecke des 3D-Ansichtsfenster). Wir fügen mit dem Menü Add » Mesh » Cube oder durch Drücken von ⇧ ShiftA » Add » Mesh » Cube einen Würfel hinzu. Durch Drücken von ⇐ Tab gelangen wir in den Edit-Modus (der Würfel wird orangefarbig angezeigt und alle Kanten und Eckpunkte (Vertices) sind hervorgehoben).



Hinzufügen eines Würfels

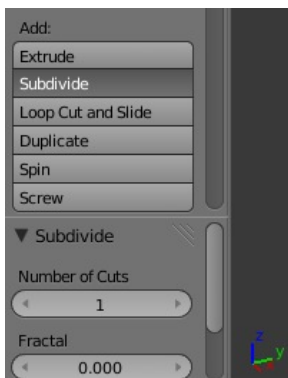
Edit Mode und Object Mode

Der Edit Mode ermöglicht es, einzelne Vertices eines Meshs zu verändern. Standardmäßig sind bei neuen Objekten alle Vertices ausgewählt (ausgewählte sind orange, nicht ausgewählte schwarz dargestellt). Im Object Mode können die Vertices nicht einzeln ausgewählt oder verändert werden, sondern das Objekt kann nur als Ganzes bearbeitet werden. Drücke ⇐ Tab, um zwischen den Modi zu wechseln. Der aktuelle Modus wird im Header-Menü (untere Menüleiste) des 3D-Ansichtsfensters angezeigt.




Gus bekommt einen Namen

Wir werden unser Lebkuchenmännlein nun offiziell "Gus" taufen. Um das zu tun, wechseln wir in den *Object*-Kontext (Siehe *Gus bekommt einen Namen*), der rechts unten seitlich im Standardlayout zu finden ist. Dort in der ersten Zeile können wir Gus einen Namen geben.



Ein Teil der Werkzeugleiste (Tool Shelf)

Unsere erste Aufgabe ist es, Gus' Körper zu erstellen, indem die Vertices des Würfels verändert werden. Die Werkzeuge hierfür finden sich in der Werkzeugleiste (Tool Shelf), die ein Teil des 3D-Ansichtsfensters ist (auf der linken Seite). Wenn die Werkzeugleiste (Tool Shelf) nicht zu sehen ist, drücke man einfach T.


Jetzt müssen wir die Subdivide (Unterteilen)-Taste in der Werkzeugleiste finden (unter Add. Diese betätigen wir einmal (*Ein Teil der Werkzeugbox (Tool Shelf)*). Dadurch wird jede Seite des Würfels in zwei Teile aufgeteilt und neue Vertices und Flächen hinzugefügt. Das Resultat wird unten dargestellt. (Um die gleiche Ansicht zu erhalten, kann man mit Num5 auch in die perspektivische Ansicht wechseln und die Ansicht durch Klicken und Ziehen mit gedrückter MMB  rotieren - nicht vergessen, Num1 (Vorderansicht) und Num5 (perspektivisch/orthographisch) zu drücken, um die vorherige Ansicht zu erhalten.)





Sobald sich der Mauszeiger im 3D-Ansichtsfenster befindet, drücken wir A, um die Auswahl aller Elemente aufzuheben. Die Vertices werden schwarz.

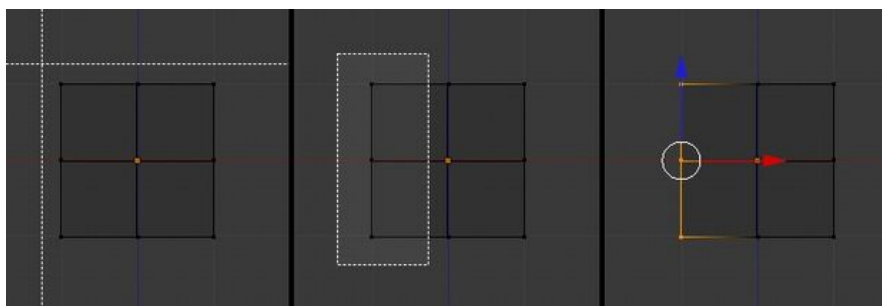
Boxauswahl – Auswahlverhalten – Nur Sichtbares Auswählen

Besonders in der orthographischen Ansicht (mit Num5 ausgewählt) können Vertices hinter anderen Vertices versteckt sein. Unser unterteilter Würfel hat 26 Vertices, jedoch sind in der orthographischen Ansicht nur neun sichtbar, weil die anderen hinter den sichtbaren liegen und versteckt sind.

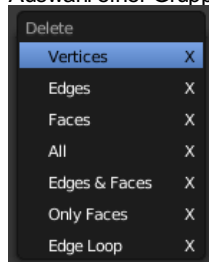
Ein RMB -Klick wählt nur einige der übereinanderliegenden Vertices, wohingegen eine Boxauswahl alle von ihnen auswählt, solange der Bounding Box (Umfang)- oder Wireframe (Gitternetz)- Anzeigemodus aktiv ist. Wenn der Ansichtsmodus Solid oder Textured ist, muss die Limit Selection to Visible- Taste deaktiviert oder nur die Darstellung von Vertices ausgewählt sein. (Die Limit Selection to Visible-Taste ist in der Header-Menüzeile des 3D-Ansichtsfensters zu finden, wenn der Solid- oder Textured-Anzeigemodus aktiv ist.)

Im weiteren Verlauf des Tutorials sollte die Taste Limit Selection to Visible  ausgeschaltet sein.

Wir drücken B, um den Boxauswahl-Modus zu aktivieren; der Cursor wird nun aus einer Reihe von rechtwinkligen grauen Linien bestehen. Den Cursor über die obere linke Ecke des Würfels bewegen, drücken und dabei die LMB  festhalten. Dann die Maus herunter und nach rechts ziehen, so dass die graue Box alle am weitesten links liegenden Vertices einschließt. Nun die LMB  loslassen (*Auswahl einer Gruppe von Vertices*).



Auswahl einer Gruppe von Vertices




Der Dialog der
Löschfunktion X

Wir drücken X und wählen Vertices aus, um die ausgewählten Vertices zu löschen (*Der Dialog der Löschfunktion X*).



Auswahlverhalten und Beschränken der Auswahl auf sichtbare Bereiche

Vor allem, wenn man sich in der orthographischen Ansicht befindet (umschaltbar mittels 5 NumPad), können möglicherweise einige Vertices hinter anderen Vertices verborgen liegen. Betrachtet man beispielsweise unseren unterteilten Würfel mit 26 Vertices im orthographischen Modus, wird man feststellen, dass nur neun von ihnen zu sehen sind, während die anderen alle

verdeckt sind. Ein Klick mit der RMB wählt nur einen dieser übereinanderliegenden Vertices aus, wobei eine Rahmenauswahl alle auswählen würde. Dies trifft zu, solange entweder der Bounding Box oder Wireframe-Ansichtsmodus aktiv ist. Ist hingegen der Solid- oder Textured-Ansichtsmodus aktiv, muss die Taste  für das Beschränken der Auswahl auf sichtbare Bereiche Limit Selection to Visible deaktiviert sein. Die benötigte Taste ist auf dem Header (untere Menüleiste) des 3D-Ansichtsfensters zu sehen, wenn entweder der Ansichtsmodus Solid oder Textured eingestellt ist.

Tipp: Circle Select (Kreisauswahl)

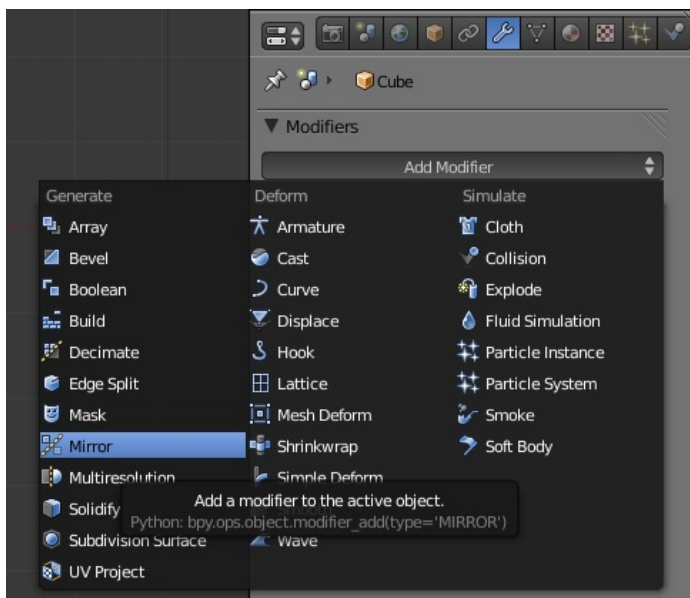
Ein anderes Auswahlwerkzeug ist Circle Select (Kreisauswahl), die durch Drücken von C aktiviert werden kann. Wenn die Kreisauswahl aktiv ist, wählt man durch Drücken oder Ziehen mit LMB Vertices aus. MMB hebt die Auswahl von Vertices auf. Drehen des Mousrades verändert die Größe des Auswahlbereiches. RMB oder ↵ Enter schließen die Auswahl ab und beenden den Kreisauswahl-Modus.

Bitte einfach etwas mit diesen Möglichkeiten herumprobieren, nachdem wir die Vertices wie oben beschrieben gelöscht haben.

Gespiegeltes Modellieren mit dem Mirror Modifier

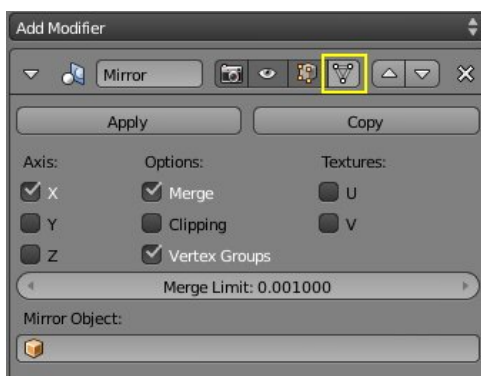
Um spiegelverkehrte Objekte zu modellieren, kann man den Mirror Modifier verwenden. Dieser erlaubt uns, nur eine Seite von Gus zu bearbeiten, während Blender die andere Seite in Echtzeit erzeugt. Geht in den Properties Editor und sucht nach dem Kontext Modifiers.

[Mehr über Modifier »](#)



Der Modifier-Kontext

Im Moment sieht es dort noch ganz schön leer aus. Wenn man die markierte Taste Add Modifier drückt, wird eine Liste geöffnet, aus der man den Mirror Modifier auswählen kann.



Taste für die Gitternetzansicht

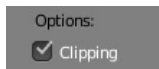
Zusätzlich zu der Möglichkeit, Objekte auf nicht-destruktive Art zu bearbeiten, erlauben Modifier eine zeitgleiche visuelle Überprüfung von Bearbeitungsschritten. In unserem Fall werden wir den Gitternetzansicht (Cage Mode) aktivieren, so dass wir im Edit Mode die gespiegelten Flächen transparent sehen können.

Daraufhin werden wir durch Drücken der Taste X, Y oder Z die Achse wählen, an der wir Gus spiegeln möchten. In diesem Fall ist dies die X-Achse.

Die Taste Merge (Zusammenführen) sorgt dafür, dass sich alle gespiegelten Vertices vereinigen, die sich in gleicher Entfernung oder innerhalb der über den Merge Limit-Schieberegler festgelegten Distanz befinden. Generell wird jeder gespiegelter Vertex, der sich

innerhalb der Limitvorgabe zur Spiegelebene befindet, mit dem entsprechenden spiegelbildlichen Vertex verschmolzen. Das Limit kann gesetzt werden zwischen **0.000** und **1.000**; wie groß man dieses Limit wählt, hängt von Art und Maßstab des jeweiligen Werkstücks ab.

Beim Modellieren von Gus würde ein Vertex, der sich mehr als **0.1** Einheiten entfernt von der Spiegelebene befindet, zu bemerken sein, aber alles, was sich näher daran befindet, würde nicht sichtbar sein. Damit in der Mitte unseres Meshes ein großes aufgerissenes Loch erscheint oder uns zwingt, einen herumvagabundierenden Vertex zu ignorieren, sollten wir das Merge Limit auf **0.1** stellen.




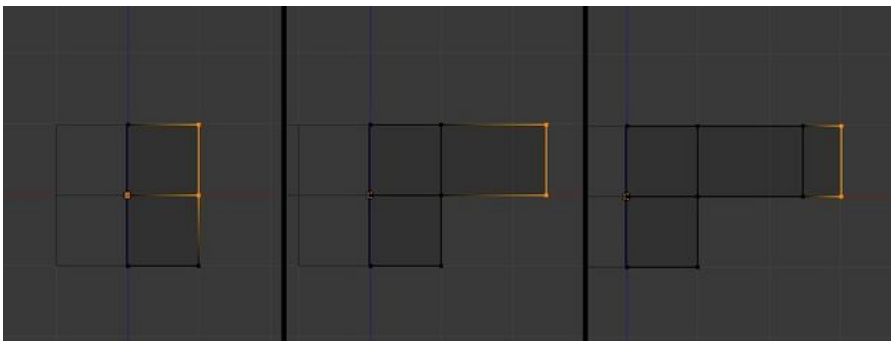
Clipping-Box

Schließlich machen wir mittels der Clipping-Box die Spiegelebene zu einer Grenze, die kein Vertex mehr überschreiten kann. Außerdem werden bei aktivem Clipping alle Vertices, die sich direkt auf der Spiegelebene befinden, dort gewissermaßen festgeklebt.

Wie Ihr also sehen könnt, macht uns der Mirror Modifier mit seinen vielen Möglichkeiten das Leben um einiges leichter.

Arme und Beine

Lasst uns nun Gus' Arme und Beine erschaffen. Wendet das gerade erlernte Vorgehen an, indem Ihr mit dem Auswahlrahmen (Box Select) die beiden am weitesten rechts befindlichen Vertices (*Extrudieren des Arms in zwei Schritten*) markiert, welche dann auch die anderen vier dahinter mit auswählen. Insgesamt sind nun also sechs Vertices aktiv. Drückt die Taste E, um sie zu extrudieren (oder benutzt die Extrude Region-Taste im Tool Shelf). Dadurch werden neue bewegliche Vertices und Faces erzeugt, die Ihr mit der Maus bewegen könnt. Bewegt sie anderthalb Blender-Einheiten nach rechts, dann klickt mit der LMB , um die Position zu fixieren. Noch einmal mit E extrudieren, um die neuen Vertices eine weitere halbe Blender-Einheit nach rechts zu bewegen. Die Abbildung unten veranschaulicht den Ablauf.



Extrudieren des Arms in zwei Schritten

Rückgängig/Wiederherstellen


Blender hat zwei Rückgängig-Funktionen, nämlich eine für den Edit Mode und die andere für den Object Mode.

Im Edit Mode drückt man StrgZ zum einmaligen Rückgängigmachen und wiederholte Male StrgZ zum mehrfachen Rückgängigmachen von Arbeitsschritten, so lange es die Kapazität des Speichers für Arbeitsschritte zulässt; ⇧ ShiftStrgZ sorgt dafür, dass vorher verworfene Änderungen doch angewendet werden. Bei Macs drückt man ⌘ Cmd statt der Taste Strg.


Zwei Dinge sollte man im Hinterkopf behalten:

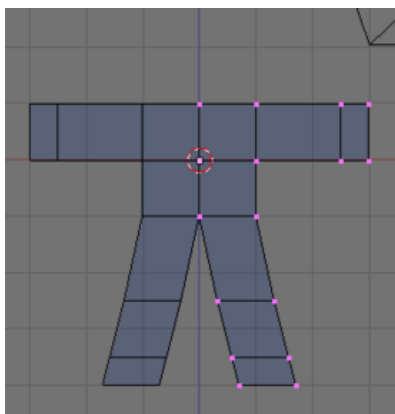
- Das Rückgängigmachen im Edit Mode funktioniert nur für das gerade ausgewählte Objekt.
- Rückgängig gemachte Arbeitsschritte sind noch nicht verloren, wenn man den Edit Mode verlässt, wohl aber, sobald man ein *anderes* Objekt im Edit Mode ausgewählt hat.

Dieselben Tastenkürzel treffen im Object Mode zu: StrgZ zum Rückgängigmachen, ⇧ ShiftStrgZ zum Wiederherstellen. Wenn man im Edit Mode Änderungen vornimmt, die für dieses Objekt noch nicht verloren sind, dann werden sie alle rückgängig gemacht in einem einzelnen Schritt mit StrgZ, sobald dieser Schritt an der Reihe ist.

Sollte man sich während einer laufenden Aktion doch noch anders entscheiden, kann man diese in vielen Fällen abbrechen und durch das Drücken von Esc oder RMB  zum Vorzustand zurückkehren.



Übereinanderliegende Vertices

Extrudieren funktioniert dergestalt, dass erst neue Vertices erzeugt und dann bewegt werden. Sollte man während des Bewegens seine Meinung ändern und die Taste Esc bzw. RMB  zum Abbrechen gedrückt haben, werden die neuen Vertices aber dennoch erhalten bleiben, und zwar direkt auf den originalen Vertices liegend! Der einfachste Weg, um zum Zustand vor dem Extrudieren zurückzukehren, ist das Rückgängigmachen mittels StrgZ. Mitunter ist es von Nutzen, wenn man auf diese Weise neue Vertices hinzufügt, um sie daraufhin bewegen (G), in ihrer Größe verändern (S) oder rotieren (R) zu können.



Körper

Gus sollte jetzt einen von Euch geschaffenen linken Arm haben (in unserer Blickrichtung) und einen rechten, von Blender hinzugefügten Arm. Wir werden das linke Bein auf dieselbe Weise bauen durch dreimaliges Extrudieren der unteren Vertices. Versucht, etwas ähnliches wie auf der Abbildung *Körper* rechts zustandezubringen. Wenn Ihr *Extrude - Region* benutzt, werdet Ihr zeitweise den Mirror Modifier mittels Abwahl der X-Option unter Axis deaktivieren müssen. Nach dem Extrudieren muss diese Einstellung wiederhergestellt werden (sonst würde Gus mit einem Rock statt mit einer Hose versehen werden).

Man kann die Bewegungseinschränkungen für die extrudierten Vertices aufheben, indem man die MMB  klickt, nachdem Ihr die Taste E und bevor Ihr die LMB  gedrückt habt. Wenn Ihr das nicht so macht, würden die Beine einfach nur gerade nach unten verlaufen, statt dass sie gleichzeitig herunter und zur Seite gezogen werden, wie es in der Abbildung *Körper* zu sehen ist.


Tipp: Zum exakten Positionieren hält man die Taste Strg gedrückt, solange man Dinge verschiebt.

Mit dem gespiegelten Modellieren sind wir nun fertig. Im nächsten Schritt werden wir mit anderen Techniken experimentieren. Wir müssen nun den rechten Teil unseres Modells *real* machen, da nichts, was man mittels Modifiern gemacht hat, permanent ist, solange man nicht ausdrücklich die Modifikationen annimmt. Wenn sich Gus also im Object Mode befindet (sollte er noch im Edit Mode sein, dann bitte jetzt die Taste \leftrightarrow Tab drücken), klickt jetzt auf die Taste *Apply* im Mirror Modifier.

Der Kopf

Gus braucht einen Kopf.



Wechselt mittels \leftrightarrow Tab in den Edit Mode.

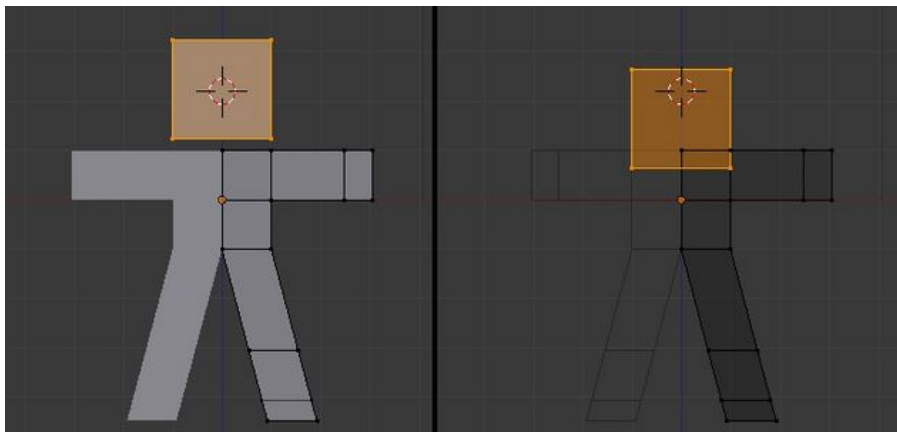
Bewegt den Mauszeiger exakt ein Quadrat oberhalb von Gus's Körper. Um den Mauszeiger an einem bestimmten Gitternetzpunkt zu positionieren, klickt mit der LMB  in die Nähe des gewünschten Ortes und drückt dann die Tastenkombination \diamond ShiftS, um das Snap Menu zum Andocken zu öffnen. Cursor to Grid platziert den Mauszeiger exakt auf einem Gitternetzpunkt. Das ist genau das, was wir gerade möchten. Cursor to Selection würde hingegen den Mauszeiger genau auf dem ausgewählten Objekt platzieren, was bisweilen auch sehr praktisch sein kann.

Fügt einen neuen Würfel (Cube) für Gus' Kopf hinzu mittels \diamond ShiftA » Add » Cube (siehe in Abbildung *Hinzufügen des Kopfes* links).

Hinzufügen von Objekten

Wenn man ein Objekt hinzufügt, während man sich im Edit Mode befindet und zuvor schon ein anderes Objekt ausgewählt hatte, wird das neue Objekt Teil dieses bereits bestehenden Objekts. Durch Hinzufügen dieses Würfels, während wir Gus' Körper noch im Edit Mode haben, wird dieser automatisch ein Teil von ihm.

Drückt jetzt G, um in den Grab Mode (Verschiebemodus) zu wechseln, und bewegt den neugeschaffenen Würfel abwärts. Man kann die Bewegung geradelinig begrenzen, indem man den Kopf nach unten bewegt und dann die MMB  klickt. Zieht Gus' neuen Kopf ungefähr ein Drittel eines Gitternetzkästchen herunter und drückt schließlich die LMB , um die Position zu fixieren (siehe in Abbildung *Hinzufügen des Kopfes* rechts).



Hinzufügen des Kopfes

Arbeiten mit unterteilten Flächen im *Subsurf* Modifier



Der Subsurf Modifier

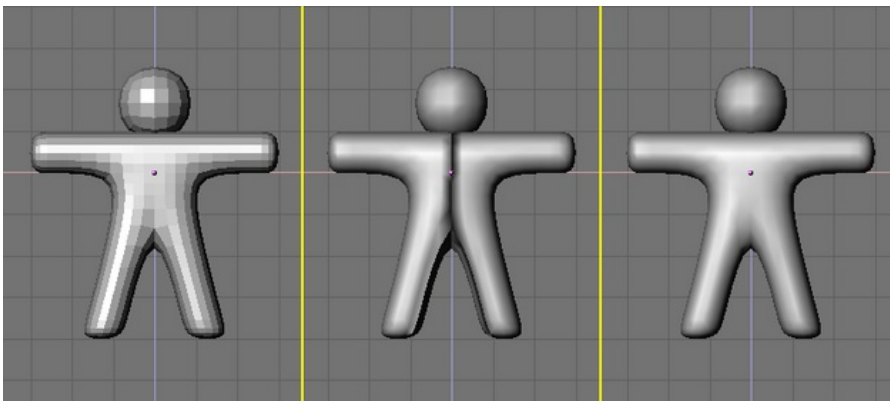
Für den nächsten Schritt werden wir den ganzen Gus auswählen müssen und nicht nur seinen Kopf (betätigt A - erforderlichenfalls zweimal).

Was wir bisher erschaffen haben, ist bestenfalls eine ganz grobe Figur. Um sie weicher zu modellieren, geht in den Modifier-Kontext und fügt einen Subdivision Surface Modifier hinzu (*Der Subsurf Modifier*). Stellt sicher, dass sowohl die Zahlentasten für View als auch für Render auf den Wert 2 oder niedriger eingestellt werden. Mit View wird der Unterteilungsgrad der Ansicht im 3D-Ansichtsfenster (3D Viewport) festgelegt; Render hingegen legt fest, wieviele Unterteilungen vom Renderer mitberücksichtigt werden sollen.

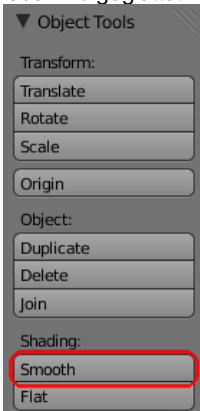
Flächenunterteilung mit SubSurfaces

SubSurfacing ist ein fortgeschrittenes Modellierungswerkzeug, das auf dynamische Weise Basis-Meshobjekte zu verfeinern hilft. Es erzeugt ein wesentlich dichteres Mesh und ordnet die Vertices in diesem feineren Mesh dergestalt an, dass sie den Konturen des originalen Basis-Meshes folgen. Die Gestalt des Objekts wird weiterhin über die Lage der Vertices des ursprünglichen Meshes bestimmt, aber die gerenderte Form ist ein feineres, glatter wirkendes Mesh.

Um Gus in Augenschein nehmen zu können, verlässt den Edit Mode mittels \leftrightarrow Tab und schaltet die Solid-Anzeige mittels Tastenkürzel Z ein. Er sollte nun so ähnlich wie in der Abbildung *Gus wird geglättet* aussehen.



Gus wird geglättet

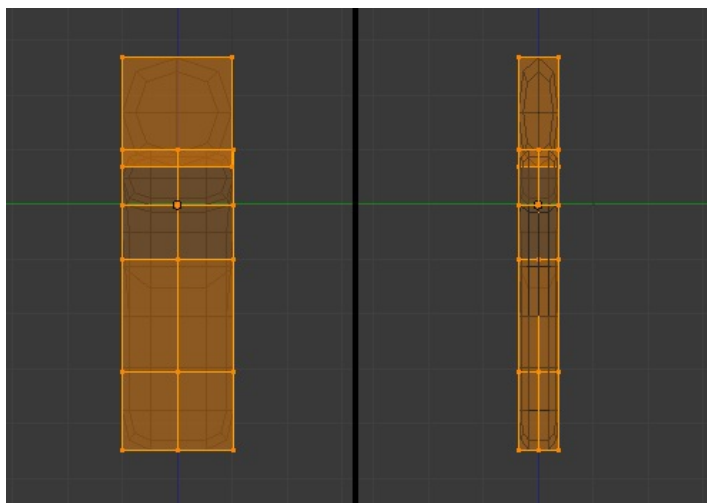


Object Tools

Um den geglätteten Gus noch etwas glatter erscheinen zu lassen, drückt auf die Taste Smooth, die sich unter der Rubrik Shading im Tool Shelf des 3D-Ansichtsfensters befindet (T). Gus wird nun ganz glatt erscheinen, wenngleich er möglicherweise in der Mitte einige seltsame schwarze Linien zeigen wird. Dies kann in der Regel dadurch vermieden werden, dass man den Mirror Modifier bereits vorher anwendet. Diese Streifen können aber auftreten, wenn Extrudieren und Wenden vor Anwendung des Mirror Modifiers vorgenommen worden sind (siehe Abbildung *Gus wird geglättet* in der Mitte). Diese Linien erscheinen, weil das durch Subsurf feinere unterteilte Mesh auf Grundlage von Daten über die Normalen-Lage des Roh-Meshes (Senkrechte Lage in bezug auf ein Face) berechnet wird. Hierbei weisen möglicherweise nicht alle Normalen in die richtige Richtung (einige Flächennormalen zeigen möglicherweise nach außen, andere hingegen nach innen). Um die Normalen zurückzusetzen, schaltet noch einmal zurück in den Edit Mode mittels (\leftrightarrow Tab), wählt alle Vertices mit (A) aus und drückt dann StrgN. Dies bewirkt eine Neuberechnung der Vertex-Normalen, damit sie alle nach außen zeigen. Jetzt sollte Gus ansprechend und glatt aussehen (siehe Abbildung *Gus wird geglättet* rechts).

Haltet nun die MMB $\left[\text{Middle Mouse Button} \right]$ gedrückt und dreht mit der Maus um Gus herum, damit Ihr ihn aus allen Blickwinkeln betrachten könnt. Doch herje, er ist zu dick!

Achsentreues Skalieren




Gus wird schlanker dank achsentreuen Skalierens



Lasst uns Guss schlanker machen:



Parameter der letzten Aktion im Tool Shelf


- Wechselt per \leftrightarrow Tab in den Edit Mode, sofern Ihr Euch nicht ohnehin schon hier befindet, und geht in den Ansichtsmodus Wireframe (Drahtgitter) mittels Z. Umschalten in die Seitenansicht mit Num3 und Auswahl aller Vertices mit A. Bei Belieben können diese Schritte auch alle im Object Mode vollzogen werden.
- Drückt S und beginnt, die Maus horizontal zu bewegen. (Klickt die MMB , um ein auf eine Achse begrenztes Skalieren zu bewirken, oder drückt Y, um zum gleichen Ergebnis zu gelangen).

Wenn Ihr nun die Maus auf Gus zubewegt, sollte er dünner werden, aber dieselbe Länge behalten.

- Auf dem Panel *Resize* des Tool Shells wird der Skalierungsfaktor angezeigt. Mit gedrückt gehaltener Strg-Taste wird der Skalierungsfaktor in einzelnen Schritten von 0.1 geändert. Skaliert Gus herunter auf den Faktor 0.2 und bestätige dieses neue Maß durch Klicken mit der LMB . Falls diese letzte Transformation schiefgelaufen sein sollte, kann man noch immer die Parameter ändern. Sie können am bereits genannten Ort betrachtet und bearbeitet werden (siehe *Parameter der letzten Aktion im Tool Shelf*).
- Rückkehr in die Frontalansicht mit Num1 und den Ansichtsmodus Solid mit Z. Dann rotieren der Ansicht mittels gedrückter MMB . Gus sieht schon viel besser aus!



Zwischenprüfung für Gus

Wir sind fast soweit, uns ein erstes Rendering anzeigen lassen zu können, aber vorher müssen wir doch noch etwas Arbeit investieren.

- Wechselt mit  Tab in den Object Mode, sofern Ihr Euch noch nicht dort befindet.

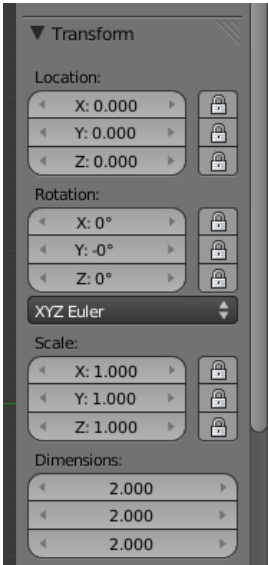


Layer (Ebenen) 1
und 10
gleichzeitig
sichtbar machen



- Klickt mit gleichzeitig gedrückten  Shift LMB  im Feld mit den Layer-Sichtbarkeitsknöpfen auf den rechts oben befindlichen im Header (untere Menüleiste) des 3D-Bearbeitungsfensters, um Layer 1 (Gus' layer) und Layer 10 (der Layer mit der Kamera und der Lampe) sichtbar zu machen (siehe *Layer (Ebenen) 1 und 10 gleichzeitig sichtbar machen*).

Tipp:



Bitte beachten, dass der zuletzt ausgewählte Layer auch der aktive Layer ist, so dass alle weiteren Ergänzungen automatisch auf Layer 10 vorgenommen werden.

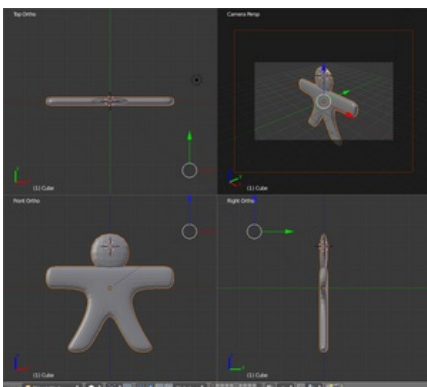


Das Transform-Panel

- Betätigt N zum Öffnen des Properties Shelves und sucht dort nach dem Transform-Panel (*Das Transform-Panel*). Die Raumposition wird durch die XYZ-Werte definiert.
- Wählt die Kamera mit RMB  aus und bewegt sie zu einer Position wie z.B. (x=7, y=-10, z=7). Dies bewerkstelligt man durch Drücken von G und Ziehen der Kamera. Möglicherweise müsst Ihr die Ansicht ein paar Male wechseln und die Kamera zum Ausrichten aller drei Koordinaten noch ein zweites Mal bewegen. Wenn Ihr es bevorzugt, numerische Werte für eine Objektposition einzugeben, dann klickt einfach mit LMB  auf einen Wert und gebt den gewünschten Wert ein.

Ausrichten der Kamera

Um die Kamera auf Gus zu richten, wählt erst die Kamera und dann Gus aus mittels  Shift RMB . Die Kamera sollte jetzt dunkelorange (ausgewählt) und Gus hellorange (ausgewählt und aktiv) dargestellt werden. Drückt jetzt StrgT und wählt aus der Dialogbox die Option TrackTo Constraint. Dadurch wird die Kamera dazu gebracht, sich immer auf Gus auszurichten und ihn zu verfolgen. Das bedeutet, dass Ihr die Kamera bewegen könnt, wie immer Ihr wollt, und trotzdem könnt Ihr sicher sein, dass Gus sich immer im Fokus der Kameraperspektive befinden wird.



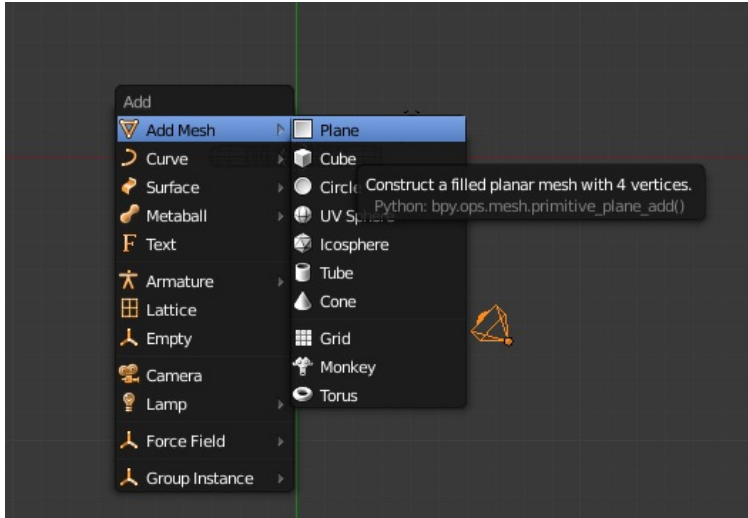
Ausrichtung der Kamera auf Gus

Die Abbildung *Ausrichtung der Kamera auf Gus* zeigt von Gus Draufsicht, Frontansicht, Seitenansicht sowie die Kameraperspektive. Um die Kameraperspektive zu erhalten, drückt 0 NumPad oder wählt die Option View » Camera. Um einen Quad View wie in der Abbildung einzustellen, drückt die Tastenkombination StrgAltQ oder wählt View » Quad View.

Die Grundfläche

Jetzt müssen wir einen Untergrund erschaffen, auf dem Gus stehen kann.

- Wechselt in der Draufsicht mittels (7 NumPad oder View » Top) und schaltet in den Object Mode um.
- Fügt eine Fläche (Plane) hinzu mit (⇧ ShiftA » Add » Mesh » Plane).



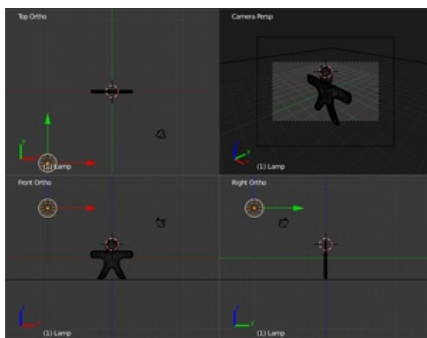
Hinweis

Es ist wichtig, zuvor den *Edit Mode* zu verlassen, da sonst jedes neu hinzugefügte Objekt Teil des gerade bearbeiteten Objekts wird, so wie dies mit Gus' Kopf der Fall war.

- Geht in die Frontansicht mit 1 NumPad oder View » Front) und bewegt nach Drücken von G die Grundfläche herunter auf Höhe von Gus' Füßen. Um sie an Gus auszurichten, verwendet die Taste Strg.
- In die Kameraperspektive wechseln mit 0 NumPad oder View » Camera) und mit weiterhin ausgewählter Fläche die Taste S betätigen, um das Skalieren beginnen zu können.
- Vergrößert die Fläche so weit, dass sich ihre Kanten außerhalb des Kamerablickwinkels befinden, was durch die helleren Bereiche in der Kameraansicht angezeigt wird.

Ausleuchtung

Lasst uns nun ein paar Lichter hinzufügen!



Einfügen einer Lichtquelle

- Bewegt in der Draufsicht (7 NumPad) die bestehende Lichtquelle (sofern Ihr noch kein Licht (Lamp) in Eurer Scene habt, könnt Ihr nun eine mit ⇧ ShiftA » Add » Lamp » Lamp vor Gus auf der kameraabgewandten Seite hinzufügen, z.B. auf Position (x= -9, y= -10, z=7) (*Einfügen einer Lichtquelle*).



Das Object Data-Icon für eine Lichtquelle

- Ist das Licht im *Object Mode* ausgewählt, geht zum Icon für den Kontext *Object Data* im Properties Editor (sieht aus wie eine kleine Sonne). Ihr werdet ein Untermenü für das Licht sehen, in dem zwischen Point, Sun, Spot, Hemi und Area gewählt werden kann. (*Das Object Data-Icon für eine Lichtquelle*).



Die Spotlight-Einstellungen

- Auf dem Panel Lamp im Properties Editor drückt auf die Auswahltaste *Spot*, um die Lichtquelle als Spotlight zu definieren (*Die Spotlight-Einstellungen*). Die blassgelbe Farbe (R=1, G=1, B=0.9) setzt man mit der weißen Taste fest, die eigentlich als Farbwähler fungiert. Einstellen der Size (Größe) unter Spot Shape (Spot-Aussehen) auf ungefähr 40 und Blend auf 1.0.
- Sorgt dafür, dass dieses Spotlight Gus verfolgt, so wie Ihr es bereits mit der Kamera tut. Dies bewerkstelligt man durch Auswahl des Spots, \diamond Shift, dann Gus und schließlich durch Drücken von CtrlT » TrackTo Constraint.
- Fügt ein zweites Licht ein, das ein gleichförmigeres Allgemeinlicht liefert, durch \diamond ShiftA » Add » Lamp » Hemi. Setzt den Wert für Energy auf 0.2 (*Hemi-Lichtquelleneinstellung*). Bewegt sie ein Stück oberhalb der Kamera (x=7, y=-10, z=9) und stellt sie wie vorher beschrieben so ein, dass sie Gus verfolgt.

Warum denn zwei Lichtquellen?

Der Gebrauch von zwei oder mehr Lichtquellen ermöglicht ein weiches, realistisches Ausleuchten, weil natürliches Licht auch in der Realität niemals nur von einem einzelnen Punkt ausgesandt wird.

Erstes Rendern



Das *Render*-Icon

Wir sind fast soweit, dass wir rendern können. Öffnet zunächst den Render-Kontext über das Icon in der Kopfzeile des Properties Editor (*Das Render-Icon*).



Der Render-Kontext

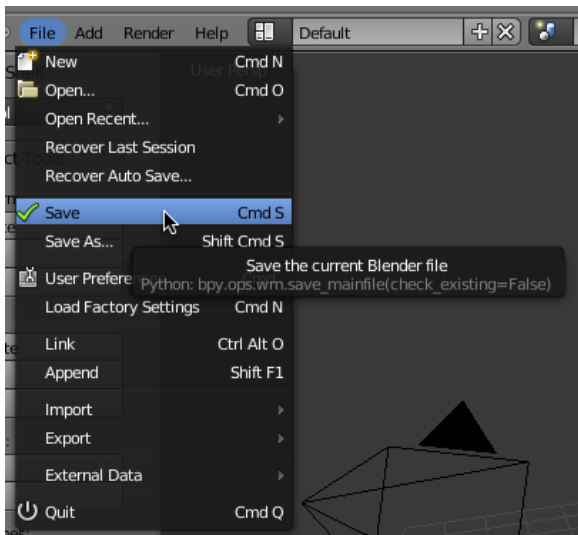
Wir werden die Standardeinstellungen so anwenden, wie sie in der Abbildung *Der Render-Kontext* gezeigt werden.

Drückt nun die Taste Image oder F12. Das Ergebnis (*Glückwunsch: Dein erstes Rendering!*) ist allerdings noch nicht sonderlich überzeugend. Wir brauchen noch Materialien, jede Menge Details wie z.B. Augen und anderes. Wenn Ihr Euch erst einmal sattgesehen habt, betätigt bitte F11, um die Rendering-Vorschau wieder verschwinden zu lassen.



Glückwunsch: Dein erstes Rendering!

Wir speichern unsere Arbeit



Das Save-Menu

Sofern Ihr es noch nicht getan habt, wäre jetzt der richtige Zeitpunkt, Eure Arbeit mittels File » Save (*Das Save-Menu*) oder CtrlS zu sichern. Blender wird Euch davor warnen, wenn Ihr versucht, eine bereits bestehende Datei zu überschreiben.

Blender speichert automatisch in Eure systemeigenes Temporärverzeichnis. Standardmäßig erfolgt dies alle 5 Minuten und der Dateiname ist eine Nummer. Diese Sicherungskopien zu laden ist ein anderer Weg, ungewollte Änderungen rückgängig zu machen.

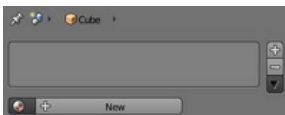
Materialien und Texturen

Nun ist es an der Zeit, Gus ein ansprechendes keksartiges Material zuzuordnen.



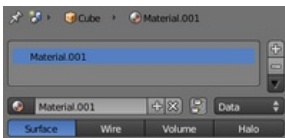
Die Taste für den Material-Kontext

- Wählt Gus aus. Drückt daraufhin im Kopf des Properties Editor auf die Taste für den Material-Kontext (*Die Taste für den Material-Kontext*), um auf die dortigen Panels zugreifen zu können.



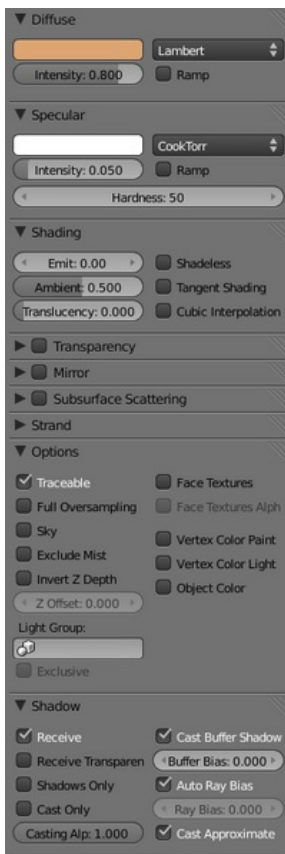
Der (fast) leere Material-Kontext

- Der Properties Editor wird erst einmal beinahe leer erscheinen, weil Gus noch kein Material besitzt. Um ein Material hinzuzufügen, klickt man auf die Taste + *New* auf dem Material-Panel (*Der (fast) leere Material-Kontext*).



Oberer Teil des ausgefüllten Material-Kontexts

- Daraufhin wird sich der Properties Editor mit Panels und Tasten füllen. Eine Eingabezeile zeigt den Materialnamen, zunächst erst einmal "Material.001", der in einem Listefeld ebenso erscheinen wird wie im Unique Datablock ID-Feld. Klickt in letzteres (unten im Oberen Teil des ausgefüllten Material-Kontexts und ändert den Materialnamen in etwas Sinnvolleres wie z.B. "GingerBread" (ohne Anführungszeichen).



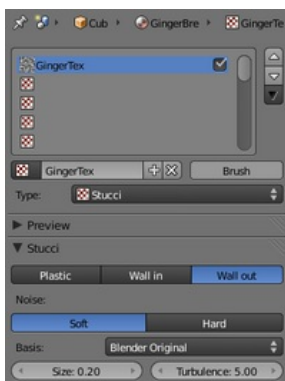
Ein erstes Lebkuchenmaterial

- Ändert die Standardwerte entsprechend der Abbildung (*Ein erstes Lebkuchenmaterial*), um ein erstes Rohmaterial zu erhalten. Bitte beachtet, dass Ihr einige Panels erst aufklappen müsst mittels LMB -Klick auf das kleine Dreieck neben der jeweiligen Kopfzeile.



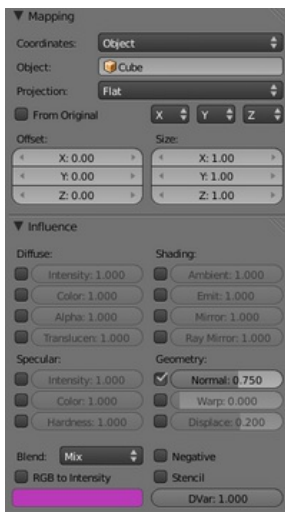
Die Taste für den Texture-Kontext

- Drückt im oberen Teil des Properties Editors auf die Taste für den Texture-Kontext (*Die Taste für den Texture-Kontext*) und wählt Add new. Damit fügen wir für den ersten Kanal eine Textur hinzu. Wir benennen sie "GingerTex."



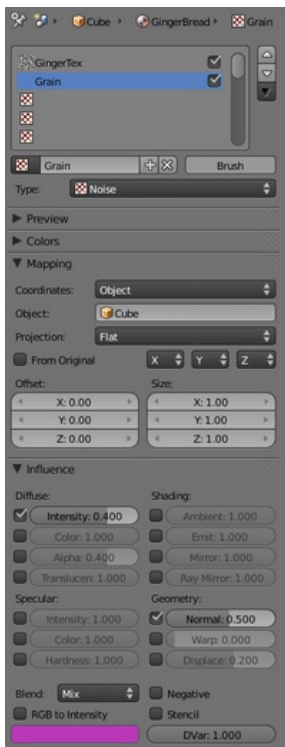
Der Texture-Kontext

- Ändert den Type von Clouds auf Stucci und stellt alle Parameter wie in der Abbildung (*The Texture context*) angegeben ein.



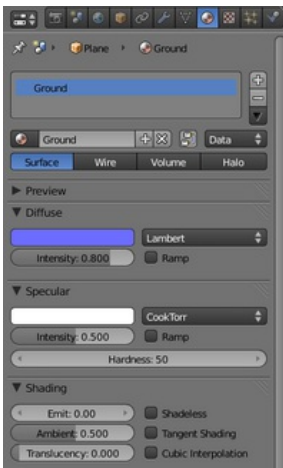
Einstellungen für die Stucci-
Textur

- Nimmt auf dem Texture-Kontext die Einstellungen auf den Panels Mapping und Influence wie in Abbildung *Einstellungen für die Stucci-Textur* vor: Entfernt den Haken aus dem Color-Kästchen und setzt dafür eines im Normal-Kästchen. Anschließend verschiebt Ihr den Normal-Regler auf **0.75**. Diese Einstellungen sorgen dafür, dass unsere Stucci-Textur als "Bumpmap" benutzt wird und Gus ein mehr biscuitartiges Aussehen verleiht.



Einstellungen für eine weitere
Noise-Textur

- Geht nun in die zweite Zeile der Texture-Liste auf dem Texture-Kontext und fügt eine weitere Textur hinzu. Benennt sie "Grain" ("Körnung") und trifft die Einstellungen gemäß Abbildung *Einstellungen für eine weitere Noise-Textur*. Die Textur an sich ist eine einfache Noise-Textur.



Ein sehr einfaches
Bodenmaterial



- Gebt dem Untergrund ein passendes Material, beispielsweise dunkelblau wie in der Abbildung *Ein sehr einfaches Bodenmaterial*. Sucht Euch den Blauton nach eigenem Geschmack aus.

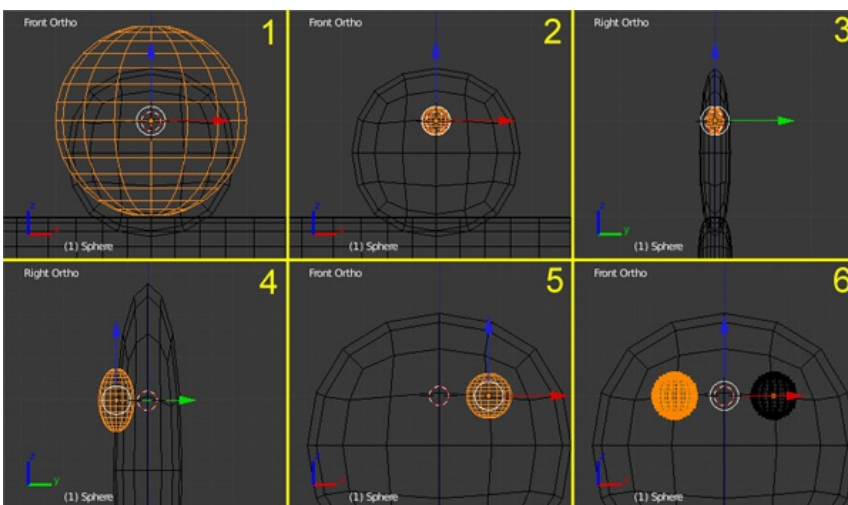
Augen und weitere Details



Um dem ganzen noch etwas Schliff zu geben, werden wir noch Augen und einige weitere Details hinzufügen.



Sichtbarkeit der
Layer

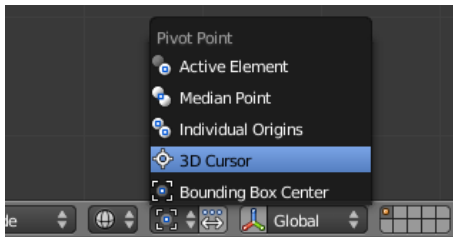
- Also erstes sorgen wir dafür, dass Layer (Ebene) 1 allein sichtbar ist. Hierzu klicken wir mit der LMB  auf das Feld für Layer 1 (*Sichtbarkeit der Layer*). Dadurch werden die Lichtquellen, die Kamera und die Grundfläche ausgeblendet.
- Platziert den Mauszeiger auf den Mittelpunkt von Gus' Kopf. (Denkt daran, dass wir uns im dreidimensionalen Raum befinden, so dass Ihr dies mindestens aus zwei Blickwinkeln begutachten könnt.)
- Fügt im Object Mode eine Kugel (Sphere) mittels  ShiftA » Add » Mesh » UVsphere. Drückt F6 und ändert die Anzahl der Segmente (Meridians) in 16. Ihr könnt das Ergebnis unter 1 in der Abbildung *Erstellen der Augen* begutachten.
- Skaliert die Kugel mit S auf einen Faktor von ungefähr **0.15** in allen Achsen herunter, dann schaltet mittels 3 NumPad in die Seitenansicht und skaliert noch einmal nur in waagerechter Ausrichtung durch zusätzliches Drücken von Y auf nunmehr **0.5** (siehe unter 2 und 3 in der Abbildung *Erstellen der Augen*).




- Falls nötig, vergrößert den Ausschnitt mittels + NumPad, Wheel  oder Ctrl MMB  und schiebt die Kugel nach Betätigen der Taste G nach links, so dass sie etwa zur Hälfte in den Kopf einsinkt (siehe unter 4 in der Abbildung *Erstellen der Augen*).
- Kehrt mit 1 NumPad in die Frontansicht zurück und bewegt die Kugel seitwärts nach rechts. Platziert sie dort, wo Gus ein Auge haben sollte (siehe unter 5 in der Abbildung *Erstellen der Augen*).

Spiegeln einer Kopie an der Position des Drehpunkts (Pivot Point)

- Wechselt in den Edit Mode (⇌ Tab). Wählt im Header des 3D-Ansichtsfensters als Drehpunkt den 3D Cursor aus (Pivot: 3D Cursor), der als Fadenkreuz mit einem Ring in der Mitte dargestellt wird. Der 3D Transform Manipulator wird daraufhin von der Kugel zur Position des Cursors springen. Alle Vertices des Auges sollten ausgewählt sein; falls nicht, dann jetzt bitte A betätigen. Drückt jetzt ⇧ ShiftD oder benutzt die Taste Duplicate auf dem Tool Shelf, um eine Kopie zu erstellen, und beendet das Platzieren der Kopie mit Esc.



Pivot: 3D Cursor

- Zum Spiegeln benutzt man StrgM, wobei zusätzlich X das Spiegeln an der X-Achse festlegt, gefolgt von LMB  oder ↵ Enter zum Bestätigen des Spiegels. Setzt daraufhin den Drehpunkt (Pivot Point) zurück auf die Standarteinstellung Median Point. Das Ergebnis ist unter 6 zu sehen in der Abbildung *Erstellen der Augen*.

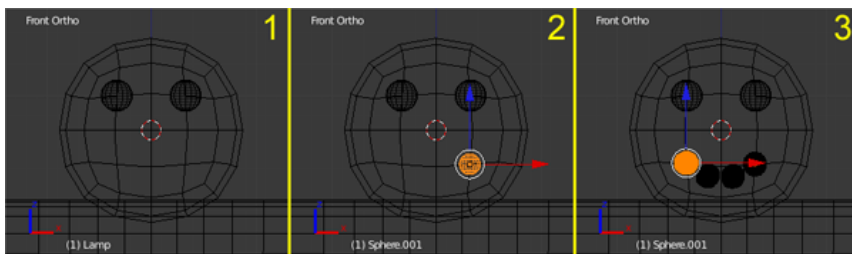
Mirroring

Spiegeln ist auch im Object Mode mittels StrgM möglich.

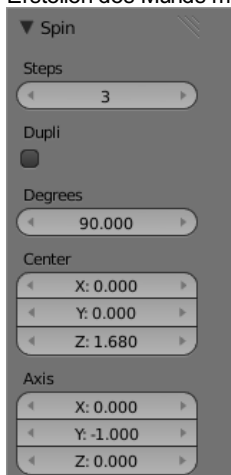
Nun hat Gus zwei Augen.

Mund


- Verlässt den Edit Mode (⇌ Tab) und setzt den Cursor so dicht wie möglich (denkt hierzu an die Taste ⇧ ShiftS) in der Mitte von Gus' Gesicht ab. Fügt eine neue Kugel (Sphere) hinzu, verkleinert und verschiebt diese genauso wie vorher die Augen, nur mit dem Unterschied, dass die allgemeine Abmessung noch kleiner sein sollen (0.1 statt 0.15). Platziert die Kugel rechts unterhalb des Cursors auf dem durch SubSurf unterteilten Mesh-Vertex, so wie unter 2 in der Abbildung *Erstellen des Mundes mit dem Spin Tool* gezeigt.



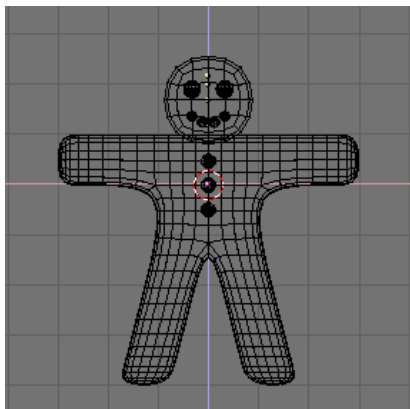
Erstellen des Mundes mit dem Spin Tool



Die Optionen für das Spin Tool auf dem Tool Shelf

- Geht in den Edit Mode (⇌ Tab). Benutzt jetzt AltR oder LMB  für Spin auf dem Tool Shelf. Mehrere Kopien der Kugel erscheinen.
- Im unteren Bereich des Tool Shelves legt die Eigenschaften für das Spin-Werkzeug fest (Drückt F6, falls diese nicht sichtbar sein

sollten, um sie in einem eigenen Popup-Fenster zu öffnen): Stellt Degrees auf **90** ein und Steps auf **3**. Das Ergebnis sollte ein Mund für Gus sein, so wie unter 3 in der Abbildung *Erstellen des Munds mit dem Spin Tool* zu sehen.



Der komplette Gus!

- Wechselt wieder in den Object Mode und fügt drei weitere Kugeln hinzu (unter dem Kopf und auf der Z-Achse angeordnet), damit Gus auch noch Knöpfe bekommt. Sobald Ihr einen Knopf gemacht habt, verlasst den Edit Mode, drückt zum Kopieren \diamond ShiftD und bewegt das Duplikat an seinen Platz, so wie in der Abbildung *Der komplette Gus!* gezeigt.

Befestigen der Kugeln

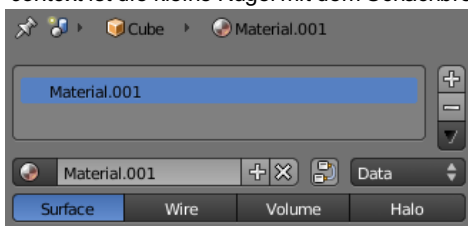
Wenn wir Gus als Ganzes fassen und bewegen möchten (was uns schon fast zum zweiten Teil dieses Tutorials führt), müssen wir die kleinen Kugeln, die Augen, Mund und Knöpfe darstellen, am Körper befestigen. Geht also in den Object Mode und drückt A, bis nichts mehr ausgewählt ist. Klickt nun eine Kugel an (falls es mehr als eine sein sollte und somit eine Gruppenauswahl zustandekommt, macht das aber auch nichts aus). Markiert mit gedrückt gehaltener \diamond Shift-Taste den Körper. Drückt daraufhin StrgP und klickt mit der linken Maustaste Object im Popup-Menü. Hebt die Markierung auf und wiederholt das Verfahren, bis alle Elemente befestigt sind.

Material für die Augen

Gebt den Augen ein schokoladenartiges Material. Der Mund sollte ein Material erhalten, das an Zuckerguss erinnert, und Gus' Knöpfe können auch ein wenig Farbe gebrauchen. Erinnert Euch daran, wie Ihr Gus selbst ein Material zugeordnet habt, und seid einfach mal etwas kreativ; in Nullkommanix wird sich Gus in letzten Schrei der Pfefferkuchenmännernmode präsentieren.

Nutzung von Materialien durch mehrere Objekte

Möchte man einem Objekt dasselbe Material wie einem anderen Objekt geben (z.B. die Wiederverwendung des weißen Materials für den Mund auf einem der Knöpfe), wählt Ihr dieses Material in der **Material Menu**-Liste aus, welche erscheint, sobald Ihr die Taste Browse ID Data neben Data Block ID Name im Material-Kontext drückt. Dieser befindet sich im Properties Editor (Im *Material context* ist die kleine Kugel mit dem Schachbrettmuster neben *Material.001* das, was Ihr braucht).



Abschließendes Rendern

Sobald Ihr das Zuordnen der Materialien abgeschlossen habt, macht Layer **10** wieder sichtbar (Könnt Ihr Euch noch erinnern, wie? Hinweis: Schaut mal in den Header des 3D-Bearbeitungsfensters), so dass auch die Lichtquellen und die Kamera wieder erscheinen. Führt mit F12 ein neues Rendern durch.

Das Ergebnis sollte mehr oder weniger wie die Abbildung (*Der komplette Gus nach dem Rendern*) ausfallen.



Der komplette Gus nach dem Rendern

Speichern

Speichert das Bild mittels F3 im UV Editor, in dem das Render-Ergebnis erscheint. Gebt den Namen des Bildes in das Dateimenü ein, wählt einen Ablageort und speichert.

Ihr könnt eines der Bilddateiformate wählen (JPEG, PNG usw.), indem Ihr es in der Spalte rechts vom Dateisuchfenster festlegt.

Blender fügt dem Dateinamen noch die entsprechende Erweiterung hinzu, woran Ihr schon gewöhnt seid (diese Funktion wurde mit V2.5 eingeführt; in V2.4x hatte man dafür noch selbst per manueller Eingabe zu sorgen).

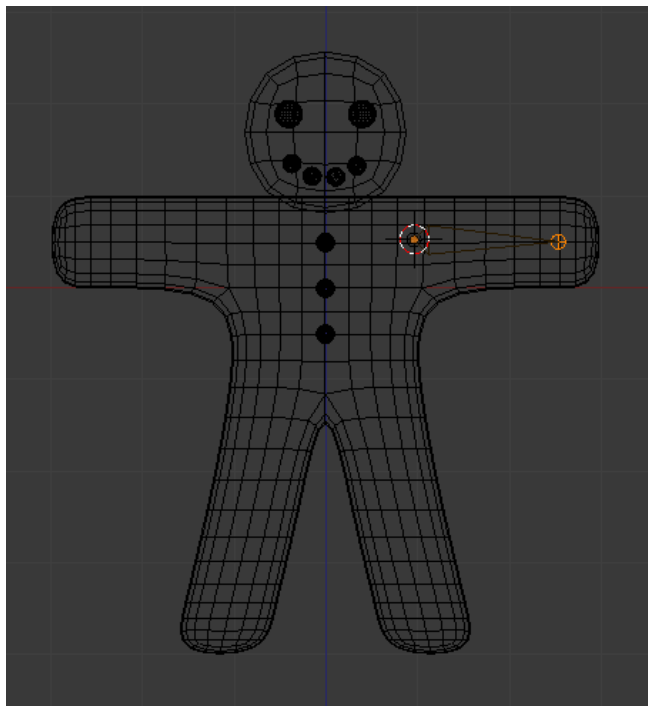
Deine erste Animation in 30+30 Minuten

Teil II: Animieren des Lebkuchenmanns

Wollten wir nur ein Standbild haben, wäre unsere Arbeit an diesem Punkt schon getan, doch wir wollen Gus bewegen! Der nächste Schritt ist also, ihm ein Skelett - auch *Armature* genannt - zu geben, mit dem er sich bewegen lässt. Dies nennt man die hohe Kunst des Riggings. Gus wird indes nur ein sehr einfaches Rigging widerfahren: Vier Gliedmaßen (zwei Arme und zwei Beine) und einige wenige Gelenke (keine Ellenbogen, lediglich Knie), und weder Füße noch Hände.

Rigging

Ein Rigging (dt. Montage oder Aufakeln) wird wie folgt durchgeführt:



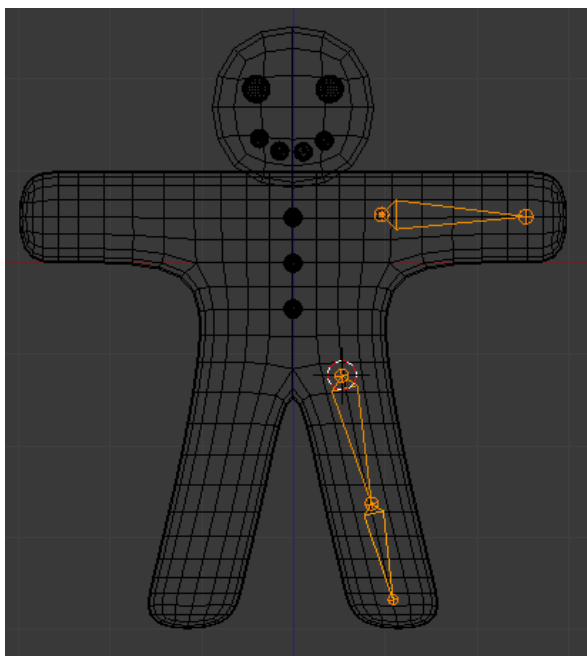
Hinzufügen des ersten Bones, eines ellenbogenlosen Arms.

- Setze im Object Mode den 3D-Cursor dorthin, wo sich Gus' Schulter befindet, und drücke \diamond ShiftA >> Add >> Armature >> Single bone. Ein rhombenförmiges Objekt wird erscheinen, das ein Bone ("Knochen") des Armature-Systems ist. Das Ende - die *Spitze* - des Bones ist ausgewählt (gelb).
- Nach dem Wechsel in den Edit Mode platziere mittels Greifen (G) die Spitze des Bones und bewege sie in Gus' Hand
- Now in Edit mode, place the tip of the bone in Gus's hand by grabbing (G) and moving it, (*Hinzufügen des ersten Bones, eines ellenbogenlosen Arms.*). Weitere Bones brauchen wir im Moment nicht. Du solltest jetzt einen Bone haben, der sich von der Schulter bis in die Handfläche erstreckt. Wenn Du die Spitze bewegst, wird Dir auffallen, dass der ganze Bone dicker wird - Du skalierst tatsächlich den gesamten Bone hoch.



Um den Prozess erquicklicher zu gestalten, spendiere bitte ab und an einen Blick auf den Lebkuchenmann und seine Armature aus verschiedenen Blickwinkeln, damit sichergestellt ist, dass sich die Armature auch tatsächlich *innerhalb* des Lebkuchenmanns befindet, wie dies auch beim menschlichen Körper der Fall ist. Das Skinning - also Überziehen mit einer Haut - würde scheitern, wenn die Bones beispielsweise vorn oder im Rücken aus dem Körper herausstünden. Die Begutachtung von vielen verschiedenen Seiten ist zudem eine allgemein übliche Technik beim Erstellen von 3D-Inhalten.

Zu den Bones-Enden

Die Enden von Bones können unterschiedliche Namen haben. In Blender werden sie gegenwärtig "head"/"tail" ("Kopf"/"Schwanz") genannt (ersterer ist das "dicke" Ende und der zweite das "schmale" Ende). Demgegenüber wurden sie früher als "root"/"tip" ("Wurzel"/"Spitze") bezeichnet, was oft als etwas weniger verwirrend empfunden wird...



Hinzufügen der zweiten und dritten Bones als Verkettung von Bones im Bein.

- Bleib im Edit Mode, dann bewege den Cursor an die Stelle, an der sich das Hüftgelenk befinden soll, und füge einen neuen Bone hinzu \diamond ShiftA.
- Greife mit (G) die gelbe Spitze und verschiebe diesen Teil des neuen Knochens in die Knieregion.
- "Verkette" nun den neuen Bone ausgehend vom Knie mit dem Fuß durch Klicken mit der Ctrl LMB  auf die Fußregion. Ein neuer verketteter Bone wird erscheinen und automatisch mit Knie und Fußende verlinkt sein (siehe *Hinzufügen der zweiten und dritten Bones als Verkettung von Bones im Bein.*). Eine andere Möglichkeit zum Verketteten des neuen Knochens wäre eine Extrusion mittels des Tastaturkürzels E. Diese Variante erzeugt den neuen Bone und platziert diesen automatisch im Grab-Modus. Zur Bestätigung der aktuellen Position der Bone-Spitze klicke man mit der LMB .

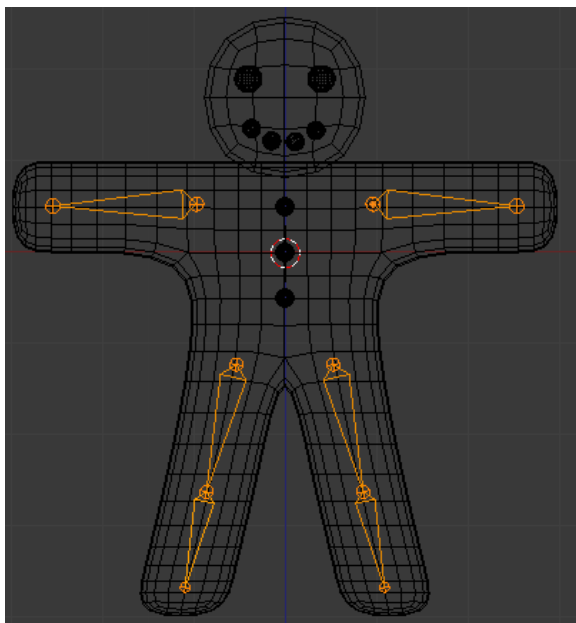
Bone-Position

Die Bones, die wir hinzufügen, werden Gus' Meshkörper verformen. Um ein ansprechendes Resultat zu erzielen, versuche eine Platzierung der Bone-Gelenke wie in der Abbildung gezeigt.

Bone Roll

Um die Bones in eine Linie zu bringen wie in (*Hinzufügen der zweiten und dritten Bones als Verkettung von Bones im Bein.*) abgebildet, musst Du eventuell das Bone Roll durch Drücken der Tastenkombination CtrlN, 3 bei ausgewählten unterem Bein ausrichten.

Wir haben nun drei Bones, die Gus' Armature bilden.

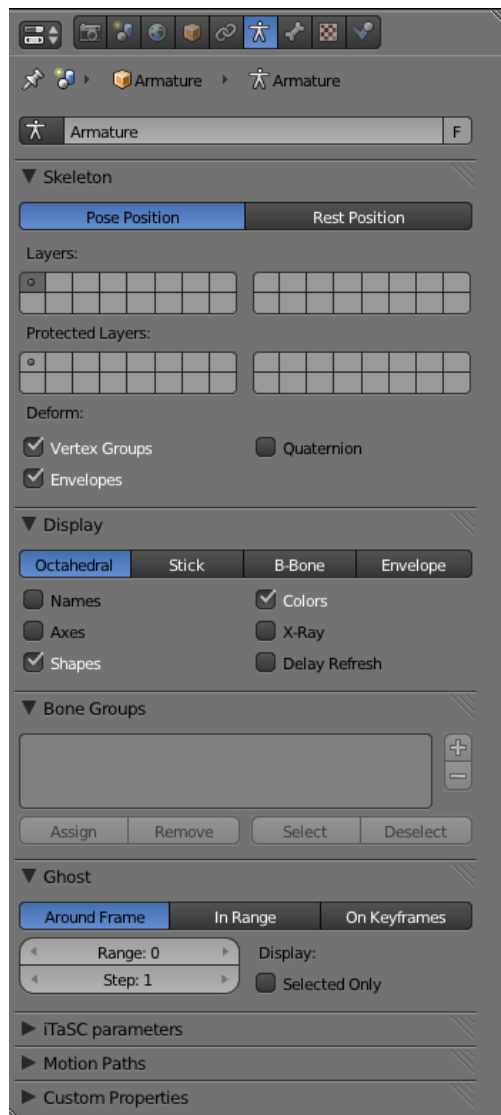


Die vollständige Armature nach Duplizieren und Wenden.

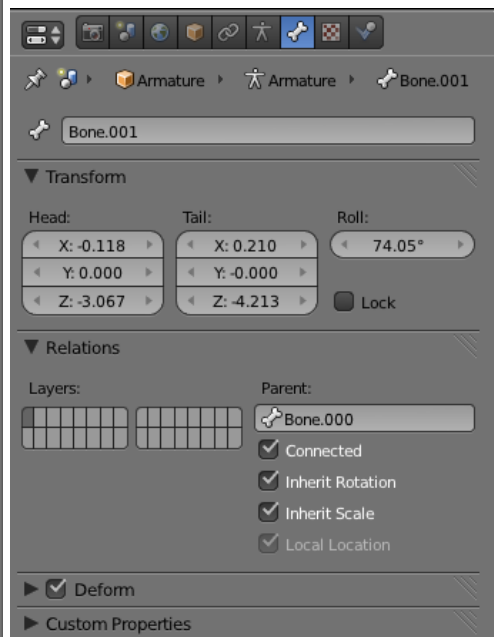
- Platziere nun den Cursor im Mittelpunkt mittels der Tastenkombination (\diamond ShiftC) und wähle alle Bones durch Drücken von A

aus. Dupliziere sie mit \diamond ShiftD und verlasse umgehend den Grab-Modus mit Esc. Stelle sicher, dass im 3D-Ansichtsfenster der Cursor als Rotations- und Skalierungszentrum eingestellt ist. Wende die duplizierten Bones entlang der X-Achse in bezug zum Cursor mittels CtrlM gefolgt von X. Zum Bestätigen der Spiegelung klicke auf die LMB . Das Ergebnis entspricht der Abbildung (*Die vollständige Armature nach Duplizieren und Umdrehen.*).

Wenn mindestens einer der Bones ausgewählt ist, zeigt das Kontextpanel Object Data die für die gesamte Armature geltenden Einstellungen an, so z.B. den Namen des Armature-Objekts und die Einstellungen für die Darstellung der Armature, wohingegen das Bone-Kontextpanel den Namen des einzelnen Bones und die auf diesen bezogenen Einstellungen enthält.



Armature- Kontextpanel



Armature-Bones-Kontextpanel

Überprüfe das Names-Kästchen auf dem (Armature-Kontextpanel sowie das Display-Panel) für die Anzeige der Namen der Bones in 3D-Ansichten, dann wähle jeden Bone aus und klicke mit der LMB auf die jeweiligen Namen der Bones im Bones-Kontext, um den Bone-Namen in etwas passenderes wie `Arm.R`, `Arm.L`, `UpLeg.R`, `LoLeg.R`, `UpLeg.L` und `LoLeg.L` zu ändern, siehe (*Armature-Bones-Kontextmenü*). Verlasse den Edit Mode über die \rightarrow Tab-Taste.

Benennen von Bones

Es ist von großer Bedeutung, die Namen mit einem angehängten ".L" oder ".R" zu versehen, um zwischen links und rechts unterscheiden zu können, so dass der Action Editor auch automatisch zwischen Deinen Posen wechseln kann.

Skinning

Jetzt müssen wir dafür sorgen, dass sich Bewegungen der Armature auf Gus's Körper auswirken: Den Vorgang nennt man Skinning (dt. häuten, bzw. auch die Hautbildung). Dabei werden Gruppen von Vertives den Bones zugewiesen sodass sie deren Bewegungen darstellen.

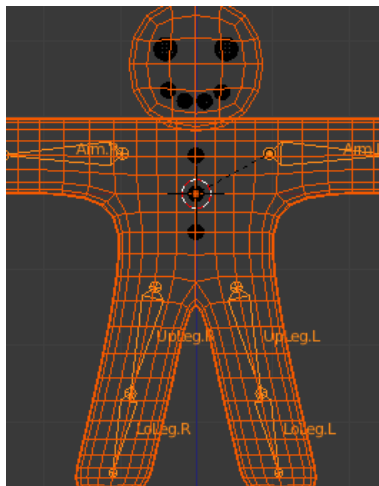
- Selektiere im Object Modus Gus's Körper, dann \diamond Shift RMB auf die Armature, sodass der Körper dunkel-orange und die Knochen hell-orange dargestellt werden.
- Dann drücke CtrlP; das Kontextmenü *Parenting menu* wird angezeigt. Darin wähle Armature Deform >> With Automatic Weights. Dadurch wird Gus's Körper zum Parent bzw. (dt. Übersetzung eher unpassend) Elternteil für die Knochen.

Vertex Gruppen, *Umhüllungen* die Reihenfolge der Modifiers

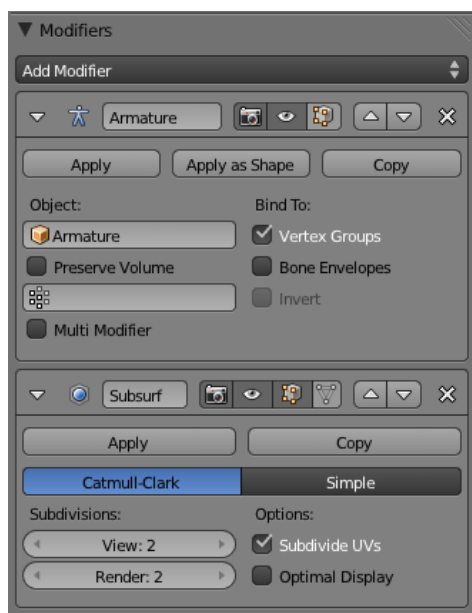
- Wenn man ein Gitternetz-Objekt mit einem Skelett verknüpft, bekommt das Gitternetz einen Armature modifier an die Liste der

Modifier angehängt. Gus's Körper hat bereits einen Subdivision Surface modifier auf der Liste. Um flüssigere Bewegungen zu erzeugen solltest Du die Reihenfolge ändern. Die Reihenfolge kann man durch einen Klick auf den hoch oder runter Pfeil im Modifier context (*The modifier stack in the Modifiers context*) ändern.

- Jeder Knochen hat Einfluß auf einen Bereich: Auf Englisch nennt man den Bereich Envelope - als deutsche Übersetzung passt *Umhüllung* am Besten. Das Skelett deformiert das Gitternetz aufgrund der zugewiesenen vertex groups UND den bone envelopes. Das kann zu ungewollten Ergebnissen führen, also deaktivieren wir in unserem Fall die Bone Envelopes am Armature Modifier (*The modifier stack in the Modifiers context*).}}



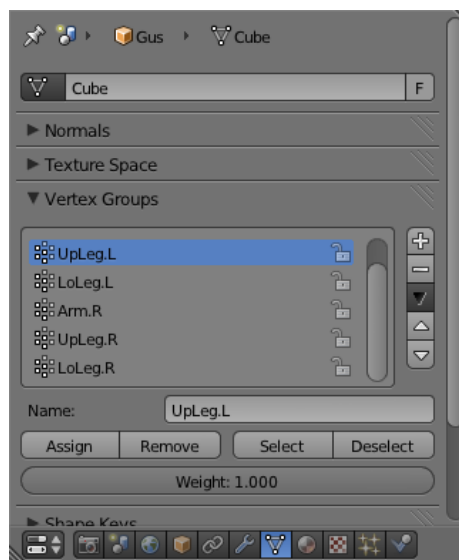
Gus und Skelett wurden *Parented*.



Der modifier stack, Armature Modifier an oberster Stelle, Bone envelope deaktiviert.

Vertex-Gruppen

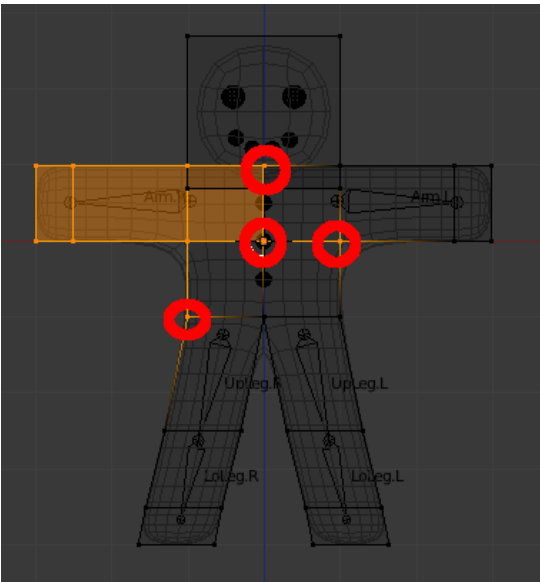
Jetzt wählt nur Gus's Körper aus und wechsele in den Edit Modus (↵ Tab). Im Object Data Kontext werden jetzt unter Vertex Groups die zuvor für die Knochen vergebenen Bezeichner aufgelistet.



Der Bereich *Vertex Groups* im Object Data Kontext.

Da wir sechs Bezeichner für Knochen bzw. Vertex Groups vergeben haben hat die Liste einen Scrollbalken erhalten. Bei einem richtig komplexen Model mit Händen und Füßen wäre sicherlich das Zehnfache an Einträgen auf der Liste! Siehe (*The vertex groups controls in the Object Data context*). Die Schaltflächen Select und Deselect (de)selektieren alle vertices der jeweiligen Gruppe.

- Deselektiere alle vertices (A. Wähle den rechten Arm (*Arm.R*) und drücke Select. Jetzt sollte Gus aussehen wie auf dem Screenshot (Gus im Edit Modus; es sind alle vertices der Gruppe *Arm.R* ausgewählt.).





Gus im Edit Modus; es sind alle vertices der Gruppe `Arm.R` ausgewählt.

- Falls Du nicht das gleiche Bild siehst hast Du wahrscheinlich die Knochen besser platziert und das auto skinning hat gleich die richtigen vertices ausgewählt.

Es ist sehr unwahrscheinlich, dass für alle Gruppen die exakt richtigen vertices zugewiesen wurden; daher werden wir die Gruppen jetzt anpassen.

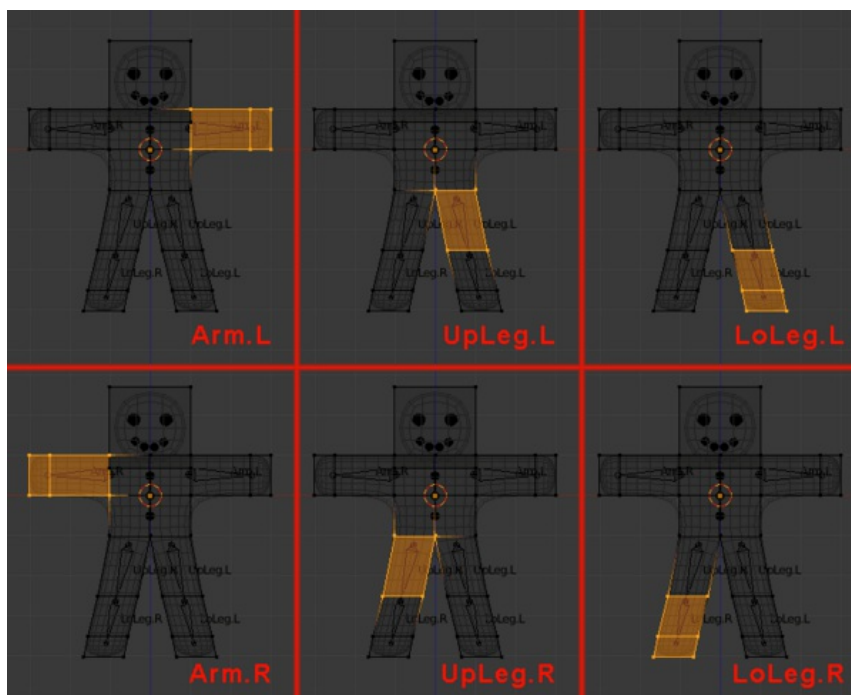
Die vertices mit den roten Kreisen in (Gus im Edit Modus; es sind alle vertices der Gruppe `Arm.R` ausgewählt.) gehören fälschlicherweise die der deformation group.

Das auto skinning war der Meinung, dass diese vertices nah genug am Knochen sind und fügte sie der deformation group hinzu. Wir möchten aber nicht, dass diese Teile von Gus' Körper von der Bewegung des rechten Arms betroffen sind.

Um die vertices aus der Gruppe zu entfernen deselektiere alle vertices, die in der Gruppe bleiben sollen. Halte \uparrow Shift und deselektiere mit RMB  oder benutze B + MMB .

Wenn nur noch die unerwünschten vertices markiert sind drücke Remove im Object Data Kontext. Dann drücke A (alles deselektieren) und selektieren die vertices von `Arm.R` erneut um das Ergebnis zu überprüfen.

So verfahren wir mit allen sechs Vertex Gruppen:



Die sechs Vertex Gruppen.

Bearbeiten der Vertex Gruppen ist auch später noch möglich

Sei sehr vorsichtig beim Pflegen der Vertex Gruppen. Wenn es später zu unerwarteten Verformungen an Gus' Körper kommt kehre zu diesem Punkt zurück!

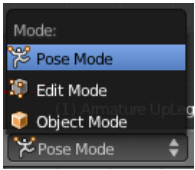
Ausblick

Unsere Deformationen werden sich nur auf Gus' Körper auswirken, nicht auf seine Augen, den Mund, die Knöpfe sowie alle

seperaten Objekte. Aber das ist kein Thema für dies simple Animation: das ist eines der komplexesten Themengebiete für große Projekte, die man durch *parenting* (-> vertices) oder durch Zusammenfügen in ein gemeinsames Gittermodell (*mesh*) lösen kann. (siehe auch [manual](#)).

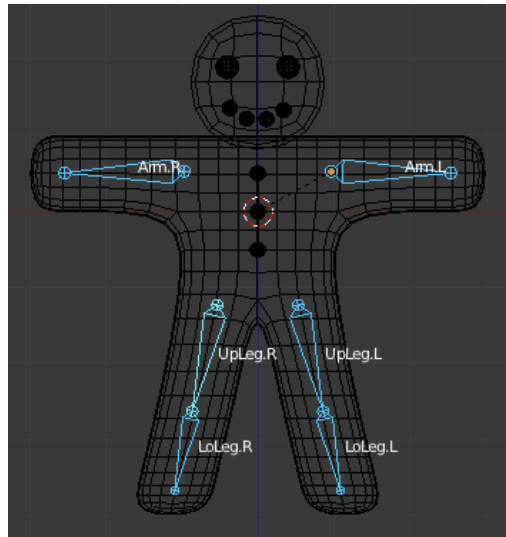
Posieren

Sobald wir einen geriggten und mit einer Skin versehenen Gus haben, können wir damit beginnen, mit ihm wie mit einer Puppe zu spielen, indem wir seine Bones bewegen und uns die Ergebnisse anschauen.



Mode-Menü im Header des 3D-Ansichtsfensters.

- Nur die Armature markieren, dann den Pose Mode aus dem Mode-Menü (*Mode menu in the 3D windowheader*) – oder einfach die Strg+Tab-Taste drücken. Diese Option ist nur verfügbar, wenn eine Armature ausgewählt ist.
- Die ausgewählten Bones werden nun blau dargestellt. Ihr befindet Euch jetzt im Pose Mode. Wenn Ihr einen Bone auswählt, bewegt ihn mit Tastenkürzel (G) oder rotiert ihn mit (R), wird der Körper dementsprechend verformt!



Ihr seid nun im Pose Mode!

Originalposition

Blender behält die Originalposition der Bones gespeichert. Ihr könnt die Armature mittels Alt+R zurücksetzen, um die Rotationen der Bones zu löschen, und Alt+G, um ihre Positionierung zu löschen. Alternativ kann man die Taste Rest Position im Object Data-Kontext benutzen, um sich temporär die Originalposition anzeigen zu lassen.

Inverse Kinematics

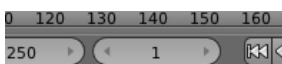
Inverse Kinematics (IK) ist die Methode, mit der man letztlich nur die Position des *letzten* Bones - oft als "*End Effector*" bezeichnet - in der Kette bestimmt. Alle anderen Bones folgen einer algorithmisch bestimmten Position, die automatisch vom *IK solver* berechnet wird, um die Kette lückenlos zu halten (d.h. IK wird die Kettengliederposition mathematisch für uns festlegen). IK erlaubt also eine viel einfachere und präzisere Positionierung von Händen und Füßen.

Forward Kinematics

While handling bones in Pose mode, notice that they act as rigid, inextensible bodies with spherical joints at the end. You can grab only the first bone of a chain and all the others will follow it. All subsequent bones in the chain cannot be grabbed and moved, you can only rotate them, so that the selected bone rotates with respect to the previous bone in the chain while all the subsequent bones of the chain follow its rotation.

This procedure, called *Forward Kinematics* (FK), is easy to follow but it makes precise location of the last bone in the chain difficult.

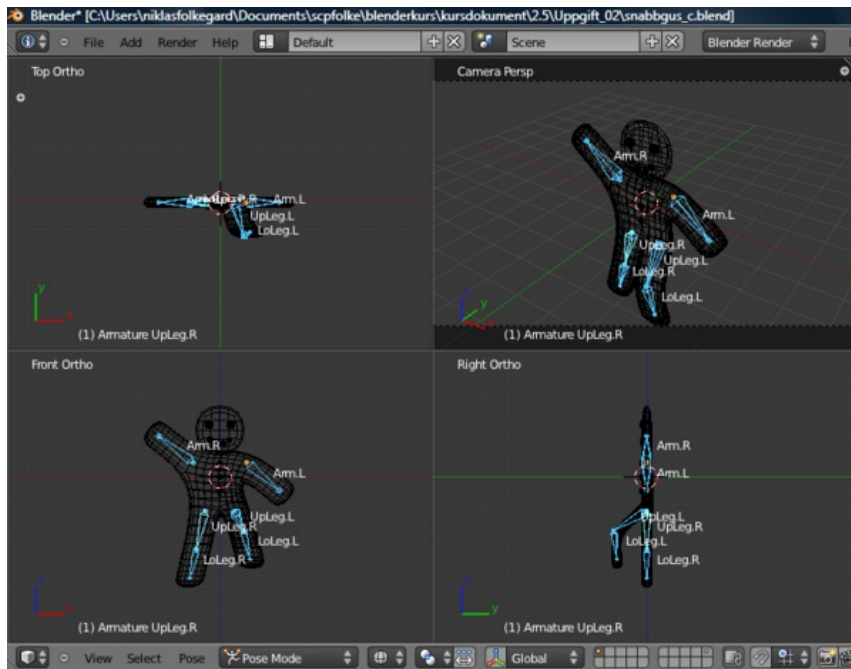
We'll make Gus walk, using FK, by defining four different poses relative to four different stages of a stride. Blender will do the work of creating a fluid animation.



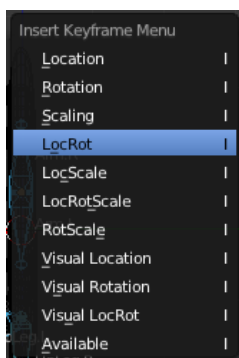
The current frame numeric field in the Timeline window

header.

- First, verify that you are at frame **1** of the timeline. The frame number appears in a numeric field in the Timeline window header (*The current frame numeric field in the Timeline windowheader*). If it is not set to **1**, set it to **1** now.
- Now, by rotating only one bone at a time (R), we'll raise `UpLeg.L` and bend `LoLeg.L` backwards while raising `Arm.R` a little and lowering `Arm.L` a little, as shown in (*Our first pose*).

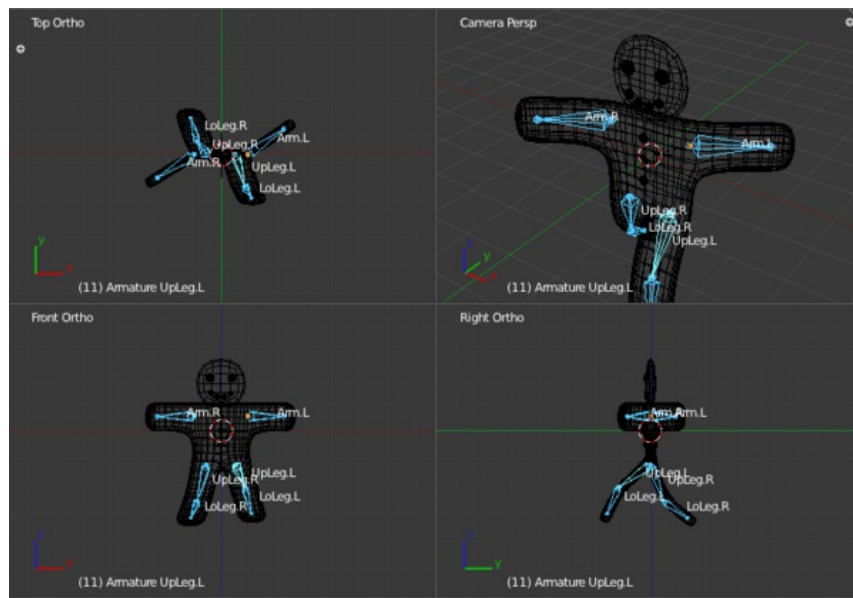


Our first pose.

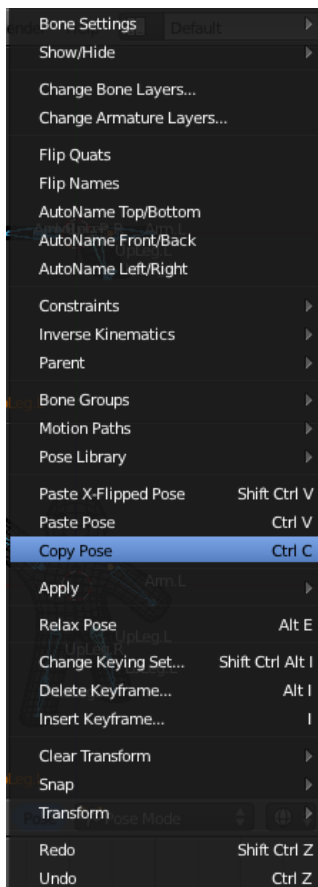


Storing the pose to the frame.

- Select all bones with A. With the mouse pointer on the 3D window, press I. A menu pops up (*Storing the pose to the frame*). Select `LocRot` from this menu. This will get the position and orientation of all bones and store them as a pose at frame **1**. This pose represents Gus in the middle of his stride, while moving his left leg forward and above the ground.
- Now move to frame **11** either by entering the number in the numeric field or by pressing `↑`. Then move Gus to a different position, like (*Our second pose*). Start with clearing the rotation on both arms using `AltR` as mentioned earlier. From the top view, rotate `Arm.R` slightly forward and `Arm.L` slightly back. Finish the pose with his left leg forward and right leg backward, both slightly bent. Gus is walking in place!



Our second pose.



Pose menu.

- Select all bones again and press I to store this pose at frame **11**, and select Rot.
- We now need a third pose at frame **21**, with the right leg up, because we are in the middle of the other half of the stride. This pose is the mirror of the one we defined at frame **1**. Therefore, return to frame **1** and, with all the bones selected, in the Pose menu of the 3D window header, select the Copy Pose entry, see (*Pose menu*). Or use CtrlC. You have now copied the current pose to the buffer.
- Go to frame **21** and paste the pose with the Paste X-Flipped Pose option in the Pose menu, see (*Pose menu*). Or use ⇧ ShiftCtrlV. This will paste the cut pose, exchanging the positions of bones with suffix “.L” with those of bones with suffix “.R”, effectively flipping it!

Bone roll

If pasting an X-flipped pose makes Gus bend in the wrong direction you may have some trouble with the Bone Roll. Select all bones in Edit Mode and press CtrlN, 3 to sort out the bone roll. Then go back to frame 1 and 11 and adjust the poses. Re-copy the pose from frame 1 and again try pasting the x-flipped pose in frame 21.

The pose is there but it has not been stored yet! You must press I » Rot with all bones selected.

- Now apply the same procedure to copy the pose at frame **11** to frame **31**, also flipping it.
- To complete the cycle, we need to copy the pose at frame **1**, *without* flipping it, to frame **41**. Do so by copying it as usual, and by

using the Paste Pose entry. Or use CtrlV. End the sequence by storing the pose with I » Rot.

Checking the animation

To preview your animation, set the current frame to **1** and press AltA in the 3D window.

Gus walks!

The single step in-place is the core of a walk, and once you have defined one there are techniques to make a character walk along a complex path. But, for the purpose of our Quick Start, this single step in-place is enough.

- In the Render context in the Properties window, set the start frame (Start) to **1** (it should already be at **1** by default) and set the end frame (End) to **40** (it is set to **250** by default, see *Settings for an animation in the Render context*). Because frame **41** is identical to frame **1**, we only need to render frames from **1** to **40** to produce the full cycle.



Settings for an animation in the Render context.

- Type `//render/` in the text field in the Output panel.
- Select AVI Raw as the file type in the Format panel. While this is generally not the best choice, mainly for file size issues, it is fast and it will run on any machine, so it suits our needs. You could also select AVI Jpeg to produce a more compact file. However, it uses lossy JPEG compression and will produce a movie that some external players might not be able to play.

Finally, press the Animation button in the Render panel. Remember that *all* the layers that you want to use in the animation must be shown! In our case, these are layers 1 and 10.

Stopping a Rendering

If you make a mistake, like forgetting to turn layer 10 on, you can stop the rendering process with Esc.

Our scene is pretty simple, and Blender will probably render each of the forty images in a few seconds. Watch them as they appear.

Stills

Of course you can always render each of your animation frames as a still by selecting the frame you wish to render and pressing the RENDER button.

Once the rendering is complete you should have a file named `0001_0040.avi` in a `render` subdirectory of your current directory – the one containing your `.blend` file. The directory can be changed from the Output panel.

You can play this file directly within Blender by pressing Play Rendered Animation in the top menu (or by using CtrlF11). The animation will automatically cycle. To stop it press Esc. We have produced only a very basic walk cycle. There is much more in Blender, as you'll soon discover reading its whole [manual](#)!

Datei Operationen

Die Optionen zum Verwalten von Dateien sind:

Datei » [Öffnen](#)

Öffnet eine Blender Datei

Datei » [Speichern](#)

Speichert die aktuelle Blender Datei

Datei » Verknüpfen or Datei » Anhängen

Sie müssen keine komplette Datei laden, bei Bedarf können Sie nur ausgewählte Teile aus einer anderen Datei importieren.

Sie die "[Verknüpfen und Anhängen](#) " Seite.

Datei » Importieren

Blender kann Informationen die auf verschiedene Weise von Formaten von Dateien gespeichert worden sind benutzen die durch andere Grafik Programme erstellt worden sind. Es tut dies durch das Ausführen eines [Skripts zum Importieren der Datei](#).

- [COLLADA](#)

Datei » Exportieren

Normalerweise speichern Sie Ihre Arbeit in einer .blend Datei, aber Sie können einige oder alle Ihrer Werke in andere Formate exportieren die von anderen Grafik Programmen verarbeitet werden können. Um dies zu tun, führen Sie ein [Export Skript](#) aus.

Dateien öffnen

Mode: Immer

Hotkey: F1

Menu: File » Open

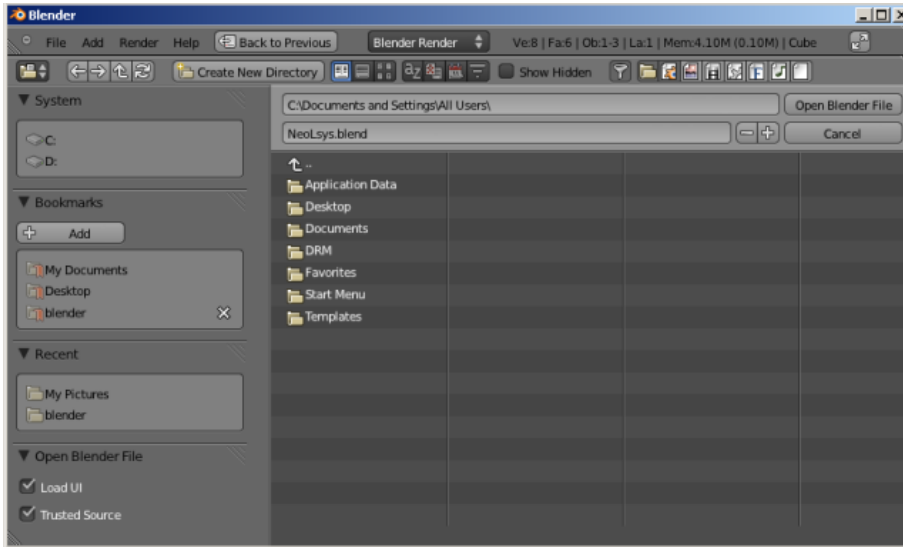
Beschreibung

Blender benutzt das `.blend`-Format um fast alles zu speichern: Objekte, Szenen, Texturen und sogar ihre Oberflächeneinstellungen.



Blender erwartet, dass Sie wissen, was Sie tun! Wenn Sie eine Datei laden, werden Sie **nicht** gefragt, ob Sie Änderungen an der geöffneten Szene speichern möchten. Das Durchführen des Ladedialogs wird als Bestätigung dafür angesehen, dass Sie wissen, was Sie tun.

Denken Sie daran Ihre Dateien zu speichern.



Benutzung des Dateimanagers und Ordernavigation

Um eine Blender-Datei von Ihrer Festplatte zu laden, drücken Sie F1. Der obige Dateimanager wird sich öffnen.



Die obere Textbox zeigt den aktuellen Dateipfad und die untere Textbox enthält den ausgewählten Dateinamen. Die P-Taste bringt Sie ins darüberliegende Verzeichnis.

Mit den + und - Tasten können Sie sich durch nummerierte Dateien bewegen, indem Sie die Nummern am Ende des Dateinamens erhöhen oder senken.

Sie können auf einen Ordner klicken um ihn zu öffnen oder auf eine `.blend`-Datei um diese zu laden.

Klicken Sie auf Cancel schließt sich der Dateimanager.

Seitenpanel

Das Panel auf der linken Seite enthält zwei Möglichkeiten Dateien zu finden, so wie weitere Optionen. Um eine Datei zu laden, wählen Sie diese mit LMB  aus und drücken Sie dann ↵ Enter oder klicken Sie auf den Open File-Button. Eine Datei kann auch geladen werden, indem Sie einfach mit MMB  auf ihren Namen klicken.

System

Das Systemmenü enthält eine Liste der verfügbaren Laufwerke zum Öffnen von Dateien. Klicken Sie auf einen Laufwerksnamen um zum Laufwerk zu springen.

Bookmarks

Es gibt sicherlich Ordner, die Sie oft benutzen wollen, ohne jedes Mal mit dem Dateimanager zum Verzeichnis zu navigieren. Um ein Verzeichnis den Lesezeichen hinzuzufügen, navigieren Sie in das Verzeichnis und klicken Sie auf Add. Um einen Ordner von der Liste zu entfernen, klicken Sie einfach auf das X-Icon daneben.

Recent

Dies ist eine Liste der zuletzt verwendeten Dateien. Sie können die Anzahl der angezeigten Dateien im File tab der user Preferences im Feld Recent Files verändern.

Optionen beim Öffnen

Blender speichert in jeder `.blend`-Datei die Einstellungen zur Benutzeroberfläche. Normalerweise wird beim Öffnen einer `.blend`-Datei die gespeicherte Benutzeroberfläche geladen. Dabei werden die Standardeinstellungen des Benutzers oder die aktuellen Einstellungen der Benutzeroberfläche überschrieben. Wenn Sie eine `.blend`-Datei mit der aktuellen Benutzeroberfläche laden wollen, schalten Sie die Funktion Load UI ab.

Das Header Panel

Das Header Panel enthält eine Reihe von Buttons um durch die Verzeichnisstruktur zu navigieren. Mit den vier Pfeilicons können Sie:

- **zum vorherigen Ordner springen**
- **das nächste Verzeichnis öffnen**
- **in das darüberliegende Verzeichnis wechseln**
- **den Inhalt des aktuellen verzeichnisses neuladen**

Sie können einen neuen Ordner im aktuellen Verzeichnis erstellen, indem Sie auf das Create New Directory Icon klicken.

Mithilfe der anderen Buttons können Sie kontrollieren, welche Dateien angezeigt und wie sie angezeigt werden sollen. Sie können die Dateien anzeigen als:

- **kurze Liste**
- **detaillierte Liste**
- **Vorschaubilder**

Sie können Dateien sortieren:

- **alphabetisch**
- **nach dem Dateityp**
- **nach dem Änderungsdatum**
- **nach der Dateigröße**

Sie können filtern, welche Dateitypen angezeigt werden sollen. Sie können die Filterfunktion aktivieren und auswählen, welche Dateitypen Sie anzeigen möchten:

- **Ordner**
- **`.blend`-Dateien**
- **Bilder**
- **Filmdateien**
- **Skripte**
- **Schriftdateien**
- **Musikdateien**
- **Textdateien**

andere Optionen beim Öffnen

Sie können auch die folgenden Werkzeuge aus dem File-Menü benutzen um Dateien zu öffnen:

Open Recent

Listet die zuletzt verwendeten Dateien auf. Sie können sie per Klick laden.

Recover Last Session

Dies lädt die `quit.blend`-Datei, welche Blender automatisch beim Schließen speichert. Es ist also die Option um die Datei der letzten Sitzung wiederherzustellen.

Recover Auto Save

Dies öffnet eine automatisch gespeicherte Datei.

Sicherheit

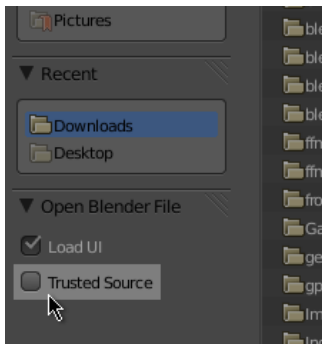
Blender hat das Ziel ein Tool auf Produktionsniveau zu sein und basiert zu einem großen Teil auf Python, einer mächtigen Skriptsprache. Python kann in Blender genutzt werden um neue Werkzeuge sowie Importer und Exporter zu erstellen und auch animation rigs steuern. Durch Python-Skripte gibt es keine Grenzen in dem, was Sie mit Blender erschaffen können.

Ein Teil der Funktionalität Pythons entsteht durch die Möglichkeit auf Ihr System zuzugreifen. Allerdings kann diese Funktion auch missbraucht werden. Es ist möglich (wenn auch nicht sehr wahrscheinlich), dass unredlich Leute `.blend`-Dateien verbreiten, welche Skripte enthalten, die Ihr System beschädigen könnten. Solche Skripte könnten als Teil eines animation rigs eingebunden sein, sodass sie ausgeführt werden, wenn eine solche `.blend`-Datei geöffnet werden sollte.



Seien Sie immer vorsichtig, wenn Sie `.blend`-Dateien oder Werkzeuge von nicht vertrauenswürdigen Quellen herunterladen!

Sicherheit



Um sich vor schädlichen `.blend`-Dateien zu schützen, ist es möglich, das Ausführen von eingebetteten Skripten beim Öffnen einer `.blend`-Datei zu verhindern. Dies hat zur Folge, dass benutzerdefinierte Werkzeuge und Python-unterstützte-rigs nicht funktionieren, es sollte jedoch kein Problem für `.blend`-Dateien wie Materialbibliotheken sein, welche diese Funktionen nicht nutzen. Es wird Ihnen immerhin die Chance geben, zu überprüfen, welche Risiken sich eventuell in einer fragwürdigen `.blend`-Datei verbergen.

Normalerweise stuft Blender alle Dateien automatisch als vertrauenswürdig ein. Wenn Sie einer Datei nicht vertrauen und Schutz möchten, können Sie die Funktion 'Trusted source' im Dateimanager unten links deaktivieren. In nicht vertrauenswürdigen `.blend`-Dateien werden eingebettete Python-Skripte beim Öffnen deaktiviert.

Dateien speichern

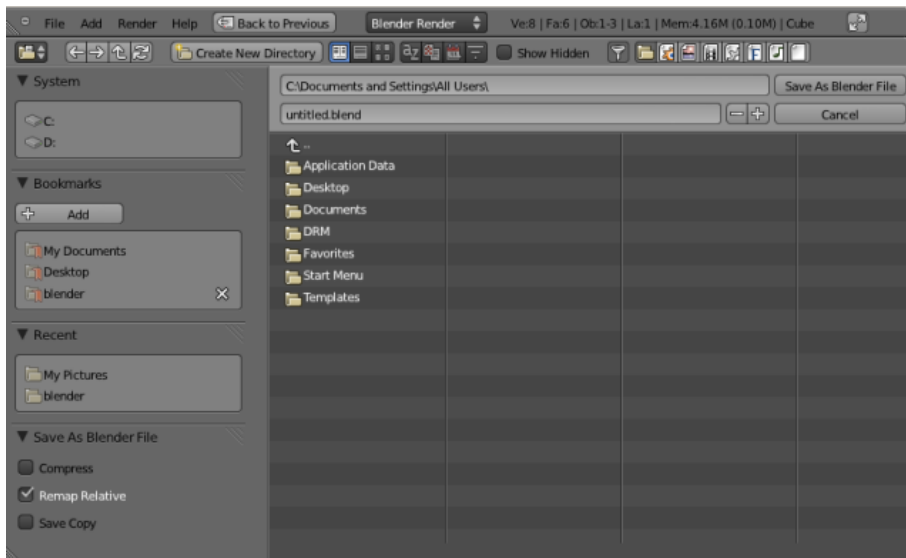
Mode: All modes

Hotkey: F2

Menu: File » Save

Beschreibung

Das Speichern von Dateien funktioniert ähnlich das Laden. Wenn Sie F2 drücken, öffnet sich der File Browser. Das Fenster sieht bis auf ein paar Optionen im Seitenpanel genauso aus wie das Ladefenster. Informationen zur Benutzung des Dateimanagers finden Sie auf der Seite [Opening files](#).



Speichern

Klicken Sie auf die untere Textbox um einen Dateinamen anzugeben. Sollte dieser nicht auf ".blend" enden, wird diese Endung automatisch hinzugefügt. Drücken Sie nun **Enter** oder klicken Sie auf **Save File** um die Datei zu speichern.

Sollte bereits eine Datei mit dem gewählten Namen existieren, werden Sie aufgefordert, das Überschreiben im **Save Over** pop-up Dialog zu bestätigen.

Abhängig von der Anzahl der automatisch zu speichernden [Versionen](#) werden alle existierenden Dateien mit dem selben Namen automatisch mit der Dateierweiterung `.blendn` versehen. Dabei ist *n* eine fortlaufende Nummer 1, 2, 3, etc. Das heißt, wenn Sie an der Datei `MyWork.blend` arbeiten und diese speichern, wird eine existierende `MyWork.blend`-Datei in `MyWork.blend1` umbenannt und Ihre neue `MyWork.blend`-Datei wird gespeichert. Dies ist eine Möglichkeit zu einer älteren Version zurückzukehren, wenn Sie massive Änderungen durchgeführt haben.

Optionen beim Speichern

Die Speicheroptionen finden Sie links unten im Seitenpanel.

Compress

Aktivieren Sie diese Optionen um große Dateien zu komprimieren.

Remap Relative

Diese Funktion aktualisiert relative Dateipfade, wenn Sie eine Datei an einem anderen Ort speichern.

Save Copy

Diese Funktion speichert eine Kopie der aktuell geöffneten Datei, ohne die Kopie aktiv zu machen.

Mehrere Versionen speichern

Der "speichern"-Dialog enthält eine praktische Funktion zum Speichern mehrerer Versionen einer Datei: Drücken Sie **+** NumPad oder **-** NumPad um eine Zahl am Ende des Dateinamens zu erhöhen oder zu verringern. Um einfach die aktuelle Datei zu überschreiben und den "speichern"-Dialog zu überspringen, drücken Sie **Ctrl+W** anstatt **F2** und bestätigen Sie die Nachfrage.

Importiere Dateien

Blender erlaubt es dir andere Dateiformate zu importieren

Collada (.dae)

[Siehe hier](#)

Motion Capture (.bvh)

Ziel

Importiert ein Objekt oder Skelett.

Skalierung

Skaliert die BVH-Datei mit diesem Wert

Start-Bild

Start-Bild für die Animation

Schleife

Wiederholt die Animations wiedergabe

Rotation

Passt die Rotation an. Benutze auch Quaternion oder Euler

Forward

Bestimmt die forward Achse

Höhe

Bestimmt die Höhen-Achse

skalierbare Vektorgrafik

Importiert eine Vektorgrafik als ein gewölbtes Objekt

Stanford (.ply)

[Siehe hier](#)

Stl (.stl)

[Siehe hier](#)

3d studio (.3ds)

[Siehe hier](#)

Größenbeschränkung

Scaliere das Modell mit 10 bis es die Größenbeschränkung erreicht hat(0 zum ausschalten)

Bildsuche

Durchsucht Unterordner nach einem dazugehörig Bild.

bestätige Transformation

Ein Workaround, wenn eine Objekt-Transformation nicht richtig importiert wurde

Forward

Bestimmt die forward Achse

Höhe

Bestimmt die Höhen-Achse

Autodesk FBX (.fbx)

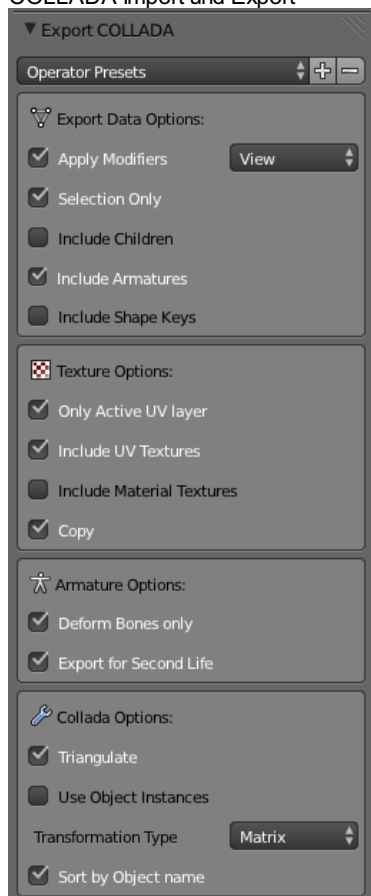
[Siehe hier](#)

Wavefront (.obj)

[Siehe hier](#)

X3D Extensible 3D (.x3d)

COLLADA Import und Export



Das Collada Modul wurde als flexibles Werkzeug zum Exportieren und Importieren von .dae Dateien implementiert. Es kümmert sich darum eine Reihe an Parametern zu bieten, welche es ermöglichen Collada Dateien von/zu mehreren Werkzeugen zu importieren/exportieren. Dennoch ist das Collada Modul noch nicht fertig gestellt, und es könnte vorkommen dass einige Szenarien noch nicht unterstützt werden.

Der Collada Exporter

Operator Vorlagen

Für Second Life Nutzer sind 2 Operator Vorlagen (siehe oben im Optionspanel) verfügbar:

- **Second Life Static**
ist nützlich zum Export von statischen Meshes.is good for exporting static meshes.
- **Second Life Rigged**
eignet sich zum Export zu Second Life Standard-Charakteren.

Besonderer Hinweis für Second Life Nutzer:

- Bitte nutzen Sie die Operator Vorlagen. Alle anderen Export Einstellungen werden nicht mit Second Life funktionieren.
- Die Charakter Orientierung muss so übereinstimmen, dass der Charakter in Richtung der positiven X-Achse sieht.
- Skalierung&Rotierung muss vor dem Export angewandt werden!

Optionen Daten-Export

Apply Modifiers

Alle aktiven Modifikatoren werden auf eine non-destruktive Art angewandt. Die Modifikatoren werden auf Kopien der Meshes angewandt. Das bedeutet dass sie die Modifikatoren vor dem Export nicht erst anwenden müssen. Dies geschieht nun automatisch im Hintergrund.

Vorschau/Render Modus

Einige Modifikatoren bieten eine Vorschau und einen Render Modus mit unterschiedlichen Mesh Einstellungen. Beim Anwenden der Modifikatoren werden beide Modi unterstützt.

Selection Only

Beim Aktivieren dieser Option wird nur die ausgewählten Objekte exportiert. Andernfalls wird die gesamte Szene mit allen sichtbaren und unsichtbaren Objekten exportiert.

Kinder einschließen

Wenn diese Option aktiviert wird, werden alle Kinder des ausgewählten Objekts unabhängig von ihrem Auswahl-Status mit exportiert.

Beispiel

Sie können jetzt nur(1) ein Skelett auswählen, dann im Exporter "include children" aktivieren, und anschließend alle Meshes die über Knochen verfügen, die an das Skelett angehängt sind, mit exportieren.

Skelett einschließen

When this option is enabled, then all armatures related to the selected objects will also be exported regardless of their selection state.

Example

You now can just select your objects, then in the exporter enable "include armatures" then the armature data is also exported.

Formschlüssel einfügen

Formschlüssel

Diese Option enthält auch die Anwendung von Formschlüsseln! Sie können damit Ihre Meshes mit ein"gebackter" Formschlüssel-Konfiguration exportieren.

Textur Optionen

Nur aktive UV Ebene

When your mesh contains multiple UV layers, then Blender exports all layers by default. This option allows you to only export the active UV layers.

Einfügen von Texturen

Blender unterstützt 2 Wege Ihre Objekte zu texturieren.

1. Durch die Nutzung von direkt zugewiesenen Oberflächentexturen
2. Durch die Nutzung von Material-basierten Bildtexturen.

While the material based image textures offer much more flexibility, using surface textures can be done very quickly without need to first render textures. Until now blender did only export material based image textures. The new option allows to directly export render results.

Textur-Export benötigt Materialien

For using surface textures, you will still have to create a material for each texture face. Then all you need to do is assign your images to the correct faces of your mesh. And finally when your object looks as you expect, just export it with "Include UV Textures". See also the "Copy" option below.

Kopieren

When you export images either material based image textures or surface textures, then we create absolute file references in the export file.

But if the "Copy" option is enabled, we will create copies of the images instead and place the copies besides the export file. In that case the file references are made relative.

Skelett Optionen

Nur Knochen deformieren

When this option is enabled, then the exporter strips all non deforming bones from the exported armatures. This option is useful when your armatures contain control bones which are not actually part of the character skeleton. For example you can now export the Avastar rig with this option enabled. The resulting exported rig is compatible to Second life. But please note the restrictions further down.

Für Second Life exportieren

Diese Option ist sehr speziell. In fact some issues with bone orientation are calculated differently when this option is enabled. This is only relevant for rigged meshes. I hope that this option will eventually be replaced by something more meaningful (and still compatible to Second Life) Hint: This option is only important when you want to export rigged meshes. For static meshes it just does nothing at all.

Collada Optionen

Triangulieren

The Mesh can be triangulated on the Fly. The triangulation is based on the same function which is used in the User interface for triangulating the current selection of faces. For full control over the triangulation you can do this manually before exporting. However this option allows to do the triangulation only for the exported data. The mesh itself is not affected.

Objektdaten

In Blender you can reuse the same mesh for multiple Objects. This is named "object instanciation". When you enable this option, then

Blender will propagate object instantiation to the Collada file.

Transformationstyp

Collada supports 2 types of Transformation matrix specifications. Either as <Matrix> or as a set of transformation decompositions (for Translate, Rotate and Scale). Note that the exporter will not strictly follow this option setting, but will rather take it as a hint to use the option if ever possible. This is so because some of the exported data types have specific rules about how the transformation matrix has to be exported. This is ongoing development and we may provide a less ambiguous method in the future.

Nach Objektnamen sortieren

The export order of data is bound to internal object order and it can not be influenced in a reliable way. this option ensures that the Geometry nodes and the Object nodes are both exported in alphabetical order.

The Collada Importer

The Collada Importer is mostly driven by the imported Data. We only have added one option for controlling the Import units:

- Import Units

If not enabled the imported data will be rescaled according to the currently used unit system. We assume that the Blender unit is 1 meter. if this option is enabled, then Blender will adjust itself to the unit system as provided by the Collada file.

Technische Details

Mesh

Import

Supported geometry types are

- tris (not tested)
- polylist
- polygons
- ngons
- trifans (not tested)
- lines

Export

Mesh data is exported as <polylist>, <lines> and <vertices>.

Licht

Import

Blender does a best effort on importing lights from a .dae. If a Blender profile is detected for lights, all values from these will be used instead. This ensures 100% reimport from a Blender exported .dae. <extra> support has been added in Blender 2.57.

Export

A Blender profile for lights has been added through the <extra> tag. The entire Lamp struct from Blender will be exported through this profile, with the exception of Light curve falloff .

Material & Effect

Export

Since Blender 2.57 some changes to export of effects have been made. Most notably <lambert> is exported if and only if specular is 0.

Animation

Export&Import

- Support for Object(Mesh, Kamera, Licht) Animationen zur Transformation. Nur euler Rotationen, welche die Standardoption für Objekte sind, werden unterstützt. For armature bone animations euler and quaternion rotation types are supported.
- Import und Export von Animationen für die folgenden Animationen werden unterstützt:-
 - Licht
 - Kamera
 - Material Effekte
- Non Skin controlling armature Knochen Animation.
- Animations of Armatures with skin deforming bones.
- Animations of Armatures in Object mode.

- Fully rigifed Armature animations. For export of rigifed Armature animations
 - Select Bake Action. (press space in 3d view and Type Bake Action)
 - If you have only the deform bones selected check "only selected". This will give smaller dae. Otherwise uncheck "Only Selected".
 - Check "Clear Constraints".
 - Bake Action.
 - Select the mesh and the deform bones. Then export to COLLADA while checking only selected option. (Selecting only the Mesh and bones is not strictly necessary. Selecting and export only selected will give smaller dae.)
 - [Demonstration](#)

Knoten

On import parent transformations for <instance_node>s is properly propagated to child node instances. Blender Materialien werden mit dem folgenden Mapping exportiert:

- Phong
- Blinn
- Labert

Für Knochen Knoten welche Knoten im Skelett Baum sind, oder falls ein Knochen mehr als ein Kind aufweist, wird ein Blender Profil mit einer <extra> Markierung versehen, und zu diesen verbundenen Knoten hinzugefügt. Dies geschieht um die Knochen->Schwanz Position bei jedem neuen Import korrekt abzuleiten.

Wichtige Dinge zum Beachten

- Objekt und Datenblock Namen sind auf 21 Zeichen (Bytes) beschränkt.
- UV Ebenen Namen sind auf 32 Zeichen (Bytes) beschränkt.
- Es wird nur die Skelett Animation auf einem Mesh mit einem einzelnen Skin Controller unterstützt
- Im Moment gibt es noch keine Unterstützung für Modifikatoren

Beim Importieren einer .dae die über <Instanz_Knote>n verfügen, wird beim Export diese Information möglicherweise verloren gehen diese Knoten werden normale <Knote>n.

Datei-Export

Sie sind nicht gezwungen an einem Projekt mit nur einer Software zu arbeiten. Sie möchten vielleicht mit Blender Ihre Arbeiten beginnen, aber eine andere Software nutzen um mit der Arbeit fortzufahren.

Blender ermöglicht dies, durch das "Exportieren" in andere Formate. Blender 2.65 unterstützt das Exportieren in 8 Formate, inklusive COLLADA.

Diese Exportmöglichkeiten sind außerdem wichtig, um Dateien mit Nutzern zu teilen, die nicht Blender, sondern beispielsweise Autodesk® 3DS Max® nutzen.

Collada (.dae)

[Siehe hier](#)

Motion Capture (.bvh)

Nur Verfügbar zum [Importieren](#)

Stanford (.ply)

[Siehe hier](#)

Apply Modifiers

Backt den Modifikator-Stapel, durch das Anwenden von allen Modifikatoren.

UV's

Export die aktiven UV Ebenen Koordinaten der Geometrie.

Normal

Export die normalen Daten der Geometrie für Glättung und harte Kanten.

Vertex Colors

Exportiert die aktive Vertex Farbebene.

Stl (.stl)

[See Here](#)

Ascii

Speichert die Datei im ASCII Dateiformat.

3d studio (.3ds)

[See Here](#)

Selection Only

Exportiert nur die ausgewählten Objekte.

Forward

gibt die Weiterleitungs Achse an.

Up

Gibt die Up Achse an.

Autodesk FBX (.fbx)

[Siehe hier](#)

Wavefront (.obj)

[Siehe hier](#)

X3D Extensible 3D (.x3d)

Unterstützte Formate

Bild Formate

Dies ist eine Liste der von Blender intern unterstützten Formate:

Grafiken mit High Dynamic Range

Blenders Input Output System unterstützt normale 32 bit Grafiken (4 x 8 Bits) oder Fließkommazahl Bilder mit 128 Bit pro Pixel (4 x 32 Bits).

Beim Lesen von High Dynamic Range Images [HDRi](#), selbst wenn sie 3 x 10 Bits groß sind, werden sie in RGBA Fließkommazahlen umgewandelt. Optional kann das z.B. im UV Bild Editor in normale 32 Bits umgewandelt werden um schnellere Darstellung zu ermöglichen.

Falls ein Bild Farben mit Fließkommazahlen verwendet, werden alle Funktionen in Blender die damit zu tun haben, darauf zurückgreifen. Das schließt den Video Sequenz Editor, das Textur Mapping, die Hintergrund Bilder, und den Compositor mit ein.

Für Hinweise zum Manipulieren oder Ansehen von HDR Bildern, sehen Sie sich den Artikel zu Kurven UI an.

- [OpenEXR \(Multiebene\)](#)
- [DPX, Cineon und Radiance HDR](#)

Andere Formate

Hier ist eine Liste sonstiger unterstützter Formate:

- BMP
- DDS
- SGI IRIS (alt!)
- PNG
- JPEG
- JPEG 2000
- Targa
- Targa Raw (nicht komprimierte Targa)
- TIFF

Film Formate

- AVI (Windows)
- AVI JPEG
- AVI Raw
- Frame Server
- H.264
- MPEG
- Ogg Theora
- QuickTime
- Xvid

Farb Modi

- BW, Bilder werden als 8 Bits in Graustufen gespeichert (nur verfügbar für PNG, JPEG, TGA, TIF)
- RGB, Bilder werden mit RGB (Farbe) gespeichert
- RGBA, Bilder werden mit RGB und Alpha Wert gespeichert (falls das unterstützt wird)

Farbtiefen

- 8 Bit Farbkanäle
- 12 Bit Farbkanäle
- 16 Bit Farbkanäle
- 32 Bit Farbkanäle

Verweiß

- [Formats](#)

OpenEXR

[ILM's OpenEXR](#) ist zum Software Industrie Standard für HDR Bilder geworden vor allem aufgrund sein flexiblen und erweiterbaren Struktur.

OpenEXR Dateien können Werte im gesamten Fließkommazahlen-Bereich speichern, einschließlich positiver, als auch negativer Zahlen.

Abgesehen vom Lesen aller OpenEXR Datei Typen mit RGBA und optionaler Z-Achse, unterstützt Blender OpenEXR auf zwei Weisen:

- [Render Ausgabe](#)
- [#Mehrfachebenen, Multi-pass, Kacheln-basierte Dateien, Kacheln-basierte Dateien](#)

Render Ausgabe

Verfügbare Optionen für die OpenEXR Render Ausgabe sind:

Half

Speichert Bilder mit Fließkommazahlen in einem besonderen 16 Bit pro Kanal Format. Das reduziert die "Bit Tiefe" zu 10 Bits mit 5 Bits Power Value und 1 Bit für eine Signatur.

Zbuf

Speichert die Tiefen Information. In Blender wird die exakte Entfernung von der Kamera in Blender Einheiten als Fließkommazahlen gespeichert.

Vorschau

Beim Animationsrendern (oder Einzelbildern über die Kommandozeile), speichert Blender das selbe Bild auch als jpg zur Vorschau oder zum Download.

Komprimierung (dieser Button befindet sich unter dem Bild Menü Button und ist standardmäßig auf "keine" eingestellt)

- PIZ, verlustfreie wavelet Komprimierung. Komprimiert ebenso Bilder mit Graustufen.
- ZIP, standard verlustfreie Kompression mit zlib
- RLE, Lauflänge codiert, verlustfrei, funktioniert gut wenn Scanlines die selben Werte haben.
- PXR24. Algorithmus mit Verlust von Pixar, konvertiert 32 Bit Fließkommazahlen, zu solchen mit 24 Bit Fließkommazahlen.

Mehrfachebenen, Multi-pass, Kacheln-basierte Dateien

Eine OpenEXR Datei kann unbegrenzt viele Ebenen und Passes enthalten die hierarchisch organisiert sind. Diese Option wird von der "Buffer speichern" Renderoption genutzt. Diese Option speichert nicht das gesamte Renderergebnis (das viele Ebenen und Passes enthalten kann) sondern speichert jedes individuelle Ergebnis in eine eigene OpenEXR Datei im standard Blender "temp" Verzeichnis.

Nach Abschluss des Renderns, und nach Freigabe von allen Render-Daten, werden diese Daten wieder zurück in den Speicher eingelesen.

DPX und Cineon

Cineon ist Kodaks standard für Filmscanning, 10 Bit pro Kanal und logarithmisch. DPX wird von Cineon abgeleitet als der ANSI/SMPTE Industrie Standard. DPX unterstützt 16 Bit Farben pro Kanal, linear sowohl als auch logarithmisch. DPX ist im moment ein weit verbreiteter Standard in der Film Hardware und Software Industrie.

DPX wie Cineon speichern nur die "sichtbaren" Farben zwischen 0.0 und 1.0 (als Ergebnis von Rendern oder Zusammenfügen). Keine Alpha Werte werden gespeichert. Der Code wurde dankenswerterweise von CinePaint kopiert. Nach CinePaints Hauptentwickler Robin Rowe greift der Code standardmäßig auf eine logarithmische Farbskala zurück. Wir haben noch nicht herausgefunden wie man das ausschaltet, speichert und liest aber ohne sichtbare Verluste.

Blender DPX Dateien (der gesamte Film Elephants Dream) wurden erfolgreich in ein Quantel IQ HD/2D finished/grading Einstellungssset importiert, also gehen wir im Moment davon aus das es für professionellen Gebrauch geeignet ist.

Radiance HDR

Radiance ist eine Software für Beleuchtungs Simulation die von Greg Ward geschrieben wurde. Da Radiance das erste (und lange Zeit einzige) HDR Bild Format hat, wird es von vielen anderen Software Paketen unterstützt.

Radiance (.hdr) Dateien speichern Farben immernoch in 8 Bit pro Komponente, aber ohne zusätzliche (geteilte) 8 Bit Exponenten Werte, die es effektiv zu 32 Bit pro Pixel machen. Es können nur RGB damit gespeichert werden.

Übersicht

Jede .blend Datei enthält eine Dateistruktur, die die gesamte Szene, Objekte, Meshes, Texturen, etc. enthält. Eine Datei kann mehrere Szenen enthalten und jede Szene ihrerseits mehrere Objekte. Objekte ihrerseits können mehrere Materialien enthalten in denen wiederum Texturen enthalten sind. Es ist möglich Verknüpfungen zwischen diesen Objekten (hier nicht im Sinne von Blender Objekten sondern Dingen) herzustellen sodass sie untergeordnete Dateien untereinander teilen können.

Diese Struktur wird im [Outliner](#) dargestellt.

Fake User

Blender löscht keine Objekte aus Szenen wie anderen Programme Teile eines Ergebnisses entfernen: bei anderen Programmen werden die Objekte die gelöscht werden sichtbar aus dem Ergebnis entfernt, Blender allerdings speichert sie nur nicht, sodass sie beim nächsten laden der Datei nicht mehr geladen werden und so im Endeffekt auch verschwinden.

Nun kann es sein das ein Teil eines Objektes schon existiert, aber das Objekt das es benutzen soll noch nicht. Blender hätte normalerweise keine Möglichkeit dieses Teilobjekt zu speichern damit es nicht verloren geht. Um das zu vermeiden gibt es in Blender die "Fake User" Funktion, sie gibt Blender die Anweisung das Teilobjekt als zugehörig zu einem vorgegebenen, falschen, nicht wirklich existierenden Benutzer Objekt zu speichern. Sie wird aktiviert mit dem F button der neben Objektnamen zu finden ist.

Wiederverwendung durch Teilen

Wie bereits erwähnt können Objekte in Blender bestimmte Unterobjekte untereinander teilen: ein Objekt kann in mehreren Szene sein, eine Farbe auf mehreren Objekten und ähnliches, ohne das es dabei nötig ist, das Objekt zu kopieren. Das ist nützlich für mehrere Dinge: Es können Szenen erstellt werden in denen nur verlinkte Objekte existieren, das ermöglicht es effektiv zusammenzuarbeiten, weil mehrere Benutzer die an der gleichen Szene arbeiten sich nicht mit versionskonflikten aufhalten müssen. Es könnte zb. ein Benutzer am Hauptcharacter, einer an der Farbgebung und einer an den Animationen arbeiten, die alle in der Endszene verlinkt sind ohne sich gegenseitig in die Quere zu kommen.

Am einfachsten ist dies vermutlich am Beispiel von [Materialien](#) zu erkennen. Es gibt einen Zähler neben geteilten Objekten der anzeigt von wievielen Objekten eine Unterdatei genutzt wird.

Siehe außerdem [wie man Objekte zwischen Szenen kopiert oder verknüpft](#).

Dateien lokal speichern

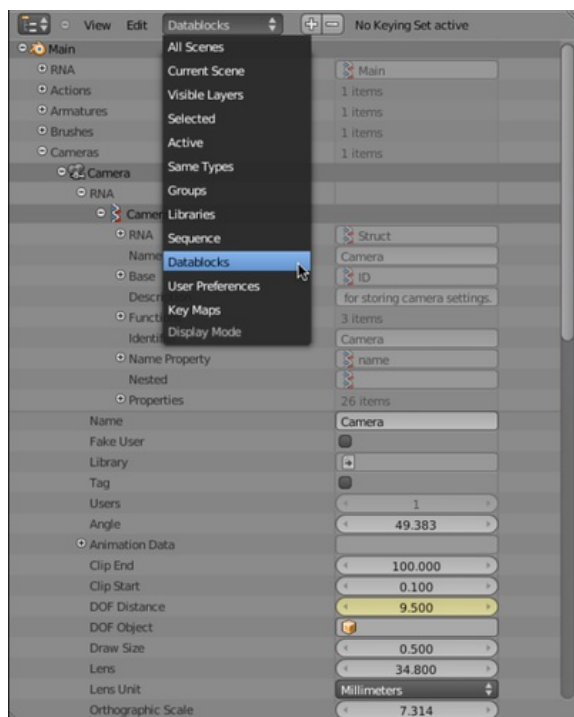
Blender kann normalerweise nur verlinkte Dateien in eine .blend datei einbinden, sodass diese dann kopiert und benutzt werden kann ohne alle Verknüpfungsdateien mitzukopieren. Zum Beispiel können Texturen die normalerweise als Bild außerhalb einer .blend vorliegen in eine solche eingeschlossen werden. Diese Option wird mit einem Button im Namensfeld links vom Namen aktiviert.

lokal gespeicherte Daten entpacken

Lokal gespeicherte Dateien können mit demselben Knopf wieder entpackt werden. Dabei gibt es die Optionen die eingepackte Datei in den gleichen Ordner wie die .blend in einem entsprechenden Ordner zu speichern, oder wenn ähnliche Dateisysteme benutzt werden an den Ort wo das Bild ursprünglich war, um es zu aktualisieren.

Datenblöcke

Die Grundeinheit für jedes Blenderprojekt, stellt der Datenblock dar. Beispiele für Datenblöcke sind: Meshes, Objekte, Materialien und Texturen. Ob es ein einfacher Kugel über einer Fläche, oder ein komplett funktionstüchtiger Film ist; Ein Projekt ist definiert durch die Datenblöcke die es enthält, die Eigenschaften die für diese gesetzt sind, und die Weise auf die die Datenblöcke miteinander verbunden sind. Datenblöcke können sich in so vielen .blend Dateien wie nötig befinden, die für eine gute Organisation des Projektes notwendig sind.



Datenblock Ansicht

Es existieren verschiedene Arten von Datenblöcken:

- RNA
- Dateiname
- Datei Hat Ungespeicherte Änderungen
- Datei ist Gespeichert
- Kameras
- Szenen
- Objekte
- Materialien
- Gruppen von Knoten
- Meshes
- Lampen
- Bibliotheken
- Bildschirme
- Fensterverwaltung
- Bilder
- Gitter
- Kurven
- Metabälle
- Vektorenschriftarten
- Texturen
- Pinsel
- Welten
- Gruppen
- Formschlüssel
- Skripte
- Texte
- Lautsprecher
- Klänge
- Skelette
- Aktionen
- Partikel
- Wachsstift
- Filme
- Masken

Kopieren und Verknüpfen von Datenblöcken

Es ist möglich jeden Typ von Datenblock zu kopieren, oder mit anderen zu verknüpfen.

Um einen Datenblock aus der Szene zu kopieren, kann die Liste Szene genutzt werden, welche sich im Kopfbereich des Fensters der Benutzereinstellungen befindet. Die Liste befindet sich zur rechten Seite der Menü und Fenster Liste. Select Neu hinzufügen um eine Kopie der aktuellen Szene anzulegen. Vollkopie kann aus der sich jetzt geöffneten Liste dazu genutzt werden um eine Kopie anzulegen. Die aktuelle Szene wird **vollständig** zur neuen Szene kopiert.

Anstelle **alles** zu kopieren, können Datenblöcke genutzt werden, indem Objekte verknüpfen ausgewählt wird, um die selben Objekt Datenblöcke zu nutzen die in die zwei anderen Szenen verknüpft sind, oder Link ObjDaten, um separate Objekte zu erzeugen welche sich die selben ObjDaten Datenblöcke (Meshes, Kurven, Materialien, usw.) teilen. Bitte beachten Sie, falls Objekte verknüpfen ausgewählt wird, kopieren Sie fast nichts, da Objekt Datenblöcke Elternteile von allen ObjDaten Datenblöcken sind: so gut wie alle Modifikationen (Objekt Position, Mesh Geometrie, ...) in einer Szene werden in den anderen wiedergespiegelt, die auch auf diese Weise verknüpft sind. ObjDaten verknüpfen erzeugt eine "reale" Kopie der Objekte, aber nicht von dessen Kinder Datenblöcken: die Positionen, Skalierungen und Rotationen der Objekte werden zu einer Szene festgelegt, aber deren Form, Materialien, Texturen, ..., werden (festgelegt durch ObjDaten Datenblöcke sein).

Kopieren und Verknüpfen von Objekt Datenblöcken

Vollkopie

- ⇧ ShiftD wird genutzt um eine normale Kopie von den ausgewählten Objekten anzulegen.

Das Objekt und einige seiner Kinder Datenblöcke werden wirklich dupliziert, die anderen Kinder werden einfach verknüpft; die zu duplizierenden Attribute können eingestellt werden in Benutzereinstellungen → Edit Methods , Knopfgruppe Duplicate with object:.

Verknüpfte Kopie

- AltD legt eine verknüpfte Kopie an.

Alle Datenblöcke des Objektes werden verknüpft.

Kopieren und Verknüpfen von anderen Datenblöcken

Wenn ein ObjDaten Datenblock von mehr als einem Objekt genutzt (verknüpft) wird, erscheint ein kleiner Knopf mit der Anzahl von verknüpften Objekten (Benutzern) in der Nähe seines Namens, in seinem Einstellungsfenster, im (Editier Kontext für Meshes, Kurven, bei Kameras, ..., im Shading Kontext, im Material unter-Kontext für Materialien, usw. Wenn Sie darauf klicken, wird eine Kopie des Datenblocks für das momentane Objekt angelegt.

Abgetrennte Datablöcke

Ein Datenblock kann abgetrennt werden. Beispielsweise wird ein Material Datenblock abgetrennt, falls das Objekt mit dem es verknüpft ist entfernt wird. Wenn ein Datenblock abgetrennt ist, wird er standardmäßig aus der .blend Datei gelöscht sobald Blender geschlossen wird. Um einen abgetrennten Datenblock in einer .blend Datei beizubehalten, klicken Sie den "F" Knopf, an der rechten Seite des Objektnamens im Objekt und Verknüpfungen Panel.

Der Status der englischen Version dieser Seite ist zur Zeit unklar.

Outliner Fenster

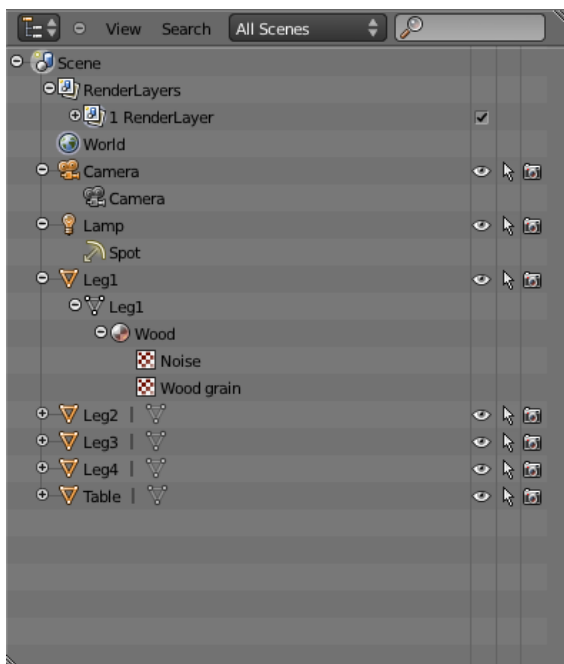
Beschreibung

Das Outliner Fenster wird dazu genutzt um einfach durch eine komplexe Szene zu navigieren. Sie bietet eine 2D Repräsentierung der komplizierten 3D Welt. Sie kann dazu genutzt werden, Dinge in Ihrer Szene ausfindig zu machen.

Nehmen Sie an, Sie niesen ausversehen während Sie gerade ein Objekt verschieben; Ihre Maus fliegt vom Schreibtisch (Gesundheit!) und das Objekt fliegt außerhalb der Bildschirmweite in den Weltraum. Sie können es dennoch leicht wiederfinden, über den Outliner. Wählen Sie es aus, und verschieben Sie es zurück in das 3D Fenster, unter den Mauszeiger mit dem Tastenkürzel (**⇧** ShiftS → Auswahl → Auswahl an Cursor).

Another more practical example is to evaluate the impact of a change on related [datablocks](#). Suppose you are looking at your `TableTop` object, and it doesn't look right, the `Wood` material doesn't look right; you want it to look more like mahogany. Since the same material can be used by many meshes, you're not sure how many things will change color when you change the material. Using the Outliner, you could find that material and trace the links that it has to every mesh in your scene.

Die Outliner Ansicht

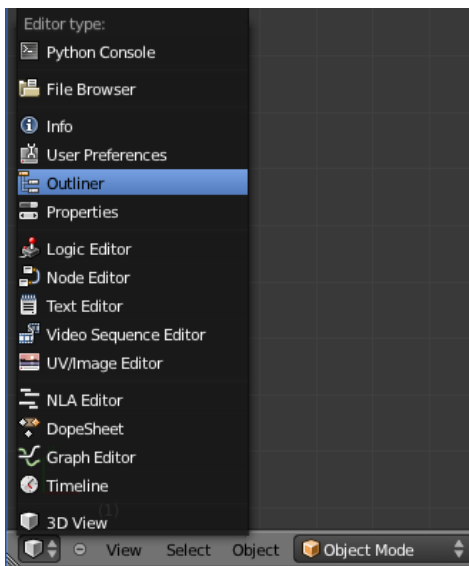


The Outliner window.



Der Outliner ist eine Art Liste, welche Dinge zu einander organisiert auflistet. Der Outliner erlaubt es:

- Die Daten der Szene anzusehen.
- Objekte der Szene aus-, und abzuwählen.
- Objekte in der Szene zu zeigen, oder zu verstecken.
- Die Auswahl zu aktivieren oder deaktivieren (um ein Objekt "unauswählbar" für die 3D Ansichten zu machen).
- Das Rendering für Objekte zu aktivieren und deaktivieren.
- Objekte aus der Szene zu entfernen.
- Verknüpfungen zu Daten aufzulösen (gleich dem Verhalten vom Drücken des X Knopf, gleich neben dem Namen des Datenblocks).
- die leichte Auswahl welche Render Ebenen tatsächlich gerendert werden sollen.
- die leichte Auswahl welcher Render Durchlauf tatsächlich gerendert werden soll (beispielsweise kann eingestellt werden, nur den Durchlauf von Glanzpunkten zu rendern).

Auswählen des Outliner Fenstertyps



Verändern des Fenstertypes zum Outliner.



Wählen Sie ein Fenster aus, und führen Sie einen LMB  auf dessen momentanen Fenstertyp Knopf aus (das ist das Symbol ganz Links in der Kopfzeile), und führen Sie einen LMB  auf Outliner aus.


Bedienung des Outliners

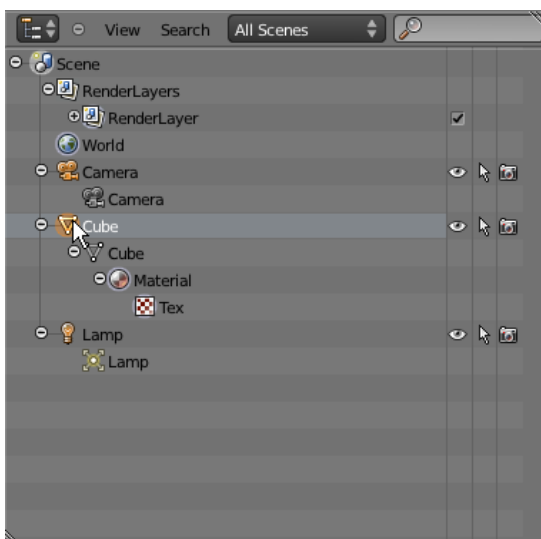
Jede Zeile im Outliner stellt einen Datenblock dar. Durch drücken des Plus-Zeichens zur linken Seite des Namens, können Sie die Ansicht des momentanen Datenblocks erweitern, und sehen welche anderen Datenblöcke er enthält. Wenn die Zeile bereits über die erweiterte Ansicht verfügt, stellt das Symbol auf der linken Seite statt einem Plus-Zeichen ein Minus-Zeichen dar. Durch klicken auf dieses Minus-Zeichens wird die erweiterte Ansicht wieder grundlegender, und die enthaltenen Objekte verborgen.

Sie können Datenblöcke im Outliner auswählen, aber das wird nicht unbedingt die Auswahl des Datenblocks in der Szene zur Folge haben. Um den Datenblock in der Szene auszuwählen, müssen Sie diese aktivieren.

Auswahl und Abwahl von Datenblöcken


Eine Einzelauswahl setzt keine Vor-Auswahl voraus; arbeiten Sie direkt mit dem LMB  (und/oder dem RMB  - Kontext-Menü, siehe unten) *mitte* im Namen/Symbol Bereich.

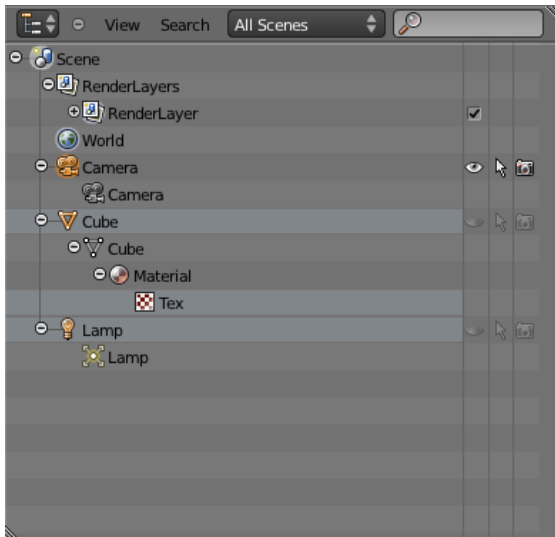
Wenn ein Objekt auf diese Weise ausgewählt wird, wird es ausgewählt und das aktive Objekt in allen anderen 3D Ansichten. Nutzen Sie diese Funktion um Objekte in Ihrer 3D Ansicht zu finden, im Outliner auszuwählen, und über das NumPad diese zu vergrößern. Falls Sie kein NumPad nutzen, rasten Sie die Ansicht ein, und zentrieren Sie den Mauszeiger auf diese über  ShiftS → Einrasten → Auswahl, und dann C.



Führen Sie einen LMB  auf die Daten des Mesh-Würfels aus um den Editormodus zu aktivieren.

Aktivieren eines Datenblocks

Aktivieren Sie den Datenblock mit einem LMB  auf das *Symbol* des Datenblock. Das Aktivieren des Datenblocks hat zur Folge, dass automatisch zum relevanten Modus gewechselt wird. Wenn beispielsweise, die Mesh-Daten des Würfels aktiviert werden, wird der Würfel ausgewählt und der Editormodus betreten, während das Aktivieren der Objektdaten des Würfels den Würfel auswählen und der Objektmodus betreten wird (siehe Rechts).



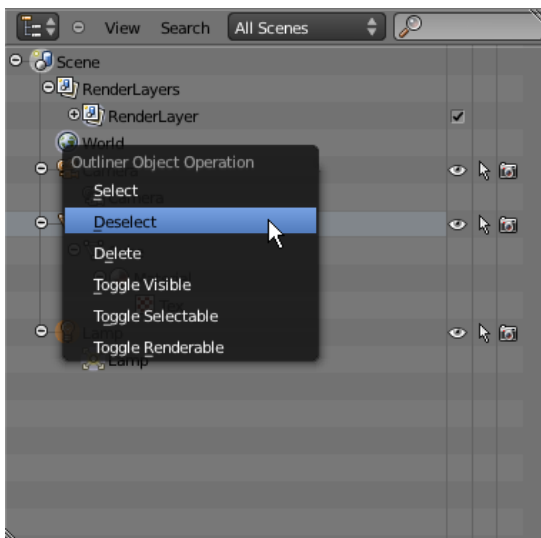
Aktivieren der Vor-Auswahl eines Datenblocks.

Wählen Sie die Vor-Auswahl einer Gruppe von Datenblöcken

Hilfreich zur Ab-, und Auswahl einer Menge an Datenblöcken. Dafür muss die Auswahl wie folgt vorbereitet werden:

- RMB oder LMB ,
- ⇧ Shift RMB oder ⇧ Shift LMB ,
- RMB und ziehen oder LMB und ziehen,

alle *außerhalb* des Namens/Symbol Bereichs. Diese Vor-Ausgewählten Datenblöcke haben eine hellere Linienfarbe. Sie können diese dann ab-, oder auswählen mit einem RMB auf den Bereich des Namens/Symbols, welcher ein Kontextmenü aufruft (siehe unten).



Kontextmenü für das Würfel Objekt.

Kontextmenü

Show the context menu for a datablock with RMB on the icon or name. Depending on the type of the pre-selected datablock(s), you will have all or part of the following options:

- Auswählen.
- Abwählt.
- Löschen.
- Verknüpfung aufheben – Zur Aufhebung der Verknüpfung von einem Datenblock zu seinem "Benutzer" (beispielsweise ein Material von seinem Mesh).
- Lokal machen – To create a "local" duplicate of this datablock.

Hinweis: Einige Datenblock-Typen haben überhaupt kein Kontextmenü!

Löschen von Datenblöcken

Nutzen Sie X zum Löschen der ausgewählten Datenblöcke.

Erweitern eines Ansicht-Level

Nutzen Sie + NumPad um ein Level tiefer in der Wurzelansicht zu gehen.

Verbergen eines Ansicht-Levels

Nutzen Sie - NumPad um ein Level höher in der Wurzelansicht zu gehen.

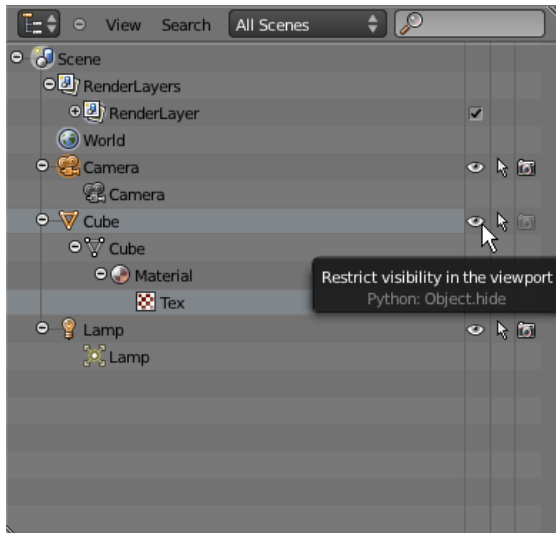
Erweitern/Verbergen von allen Ansicht-Leveln

Nutzen Sie A um alle Ansicht-Level in der Wurzelansicht erweitert anzuzeigen, oder zu verbergen.

Wechsel von Objekt-Level Beschränkungen

Die drei folgenden Optionen, auf der rechten Seite des Outliner Fensters, sind nur für Objekte verfügbar:

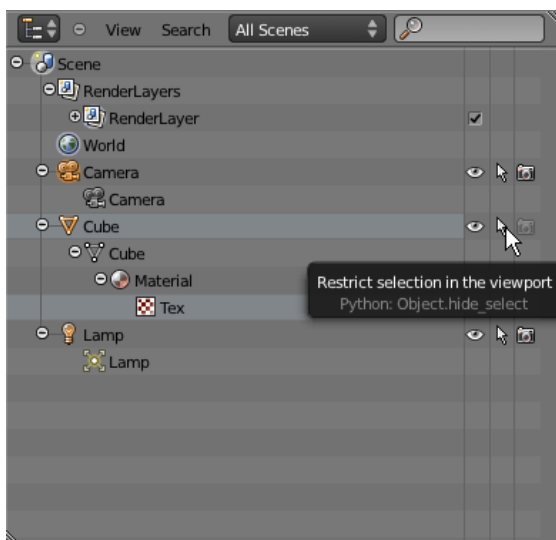
Sichtbar



Schränkt die Sichtbarkeit ein

Toggle visibility by clicking the “eye” icon for the object on the right-hand side of the Outliner. Useful for complex scenes when you don't want to assign the object to another layer. This will only work on visible layers - an object on an invisible layer will still be invisible regardless of what the Outliner says. V will toggle this property for any objects that are pre-selected in the Outliner.

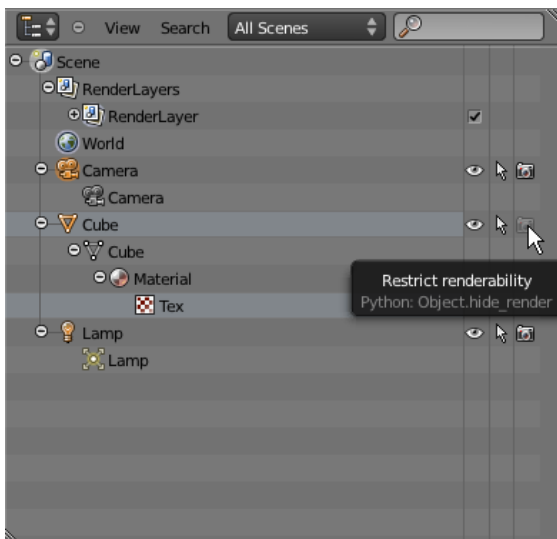
Auswählbar



Restrict selection

Toggle selectability by clicking the “arrow” icon. This is useful for if you have placed something in the scene and don't want to accidentally select it when working on something else. S will toggle this property for any objects that are pre-selected in the Outliner.

Renderbar



Restrict renderability

Toggle rendering by clicking the “camera” icon. This will still keep the object visible in the scene, but it will be ignored by the renderer. R will toggle this property for any objects that are pre-selected in the Outliner.

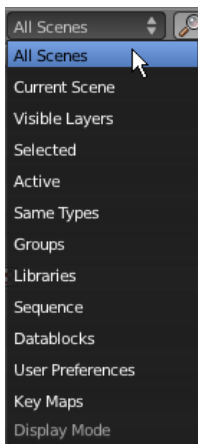
Suchen

You can search the file for datablocks, either by using the Search menu in the header of the Outliner, or by using one of the following hotkeys:

- F - Find.
- CtrlF - Find (case sensitive).
- AltF - Find complete.
- CtrlAltF - Find complete (case sensitive).
- ⇧ ShiftF - Find again.

Matching datablocks will be automatically selected.

Filtern der Ansicht

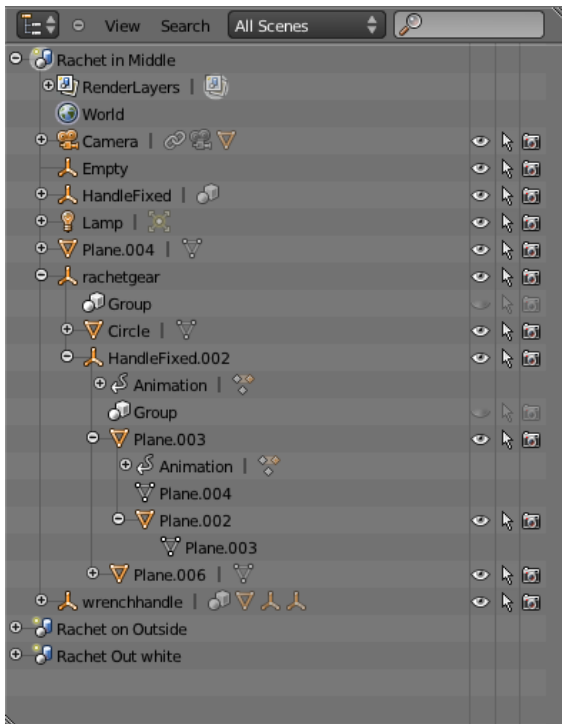


Outliner Ansicht
Dropdown.

The window header has a field to let you select what the outliner should show in the outline. By default, the outliner shows All Scenes. You can select to show only the current scene, datablocks that have been selected, objects that are on currently selected layers, etc. These selects are to help you *narrow the list* of objects so that you can find things quickly and easily.

- All Scenes - Shows *everything* the outliner can display (in all scenes, all layers, etc.)
- Current Scene - Shows everything in the current scene.
- Visible Layers - Shows everything on the visible (currently selected) layers in the current scene. Use the [layers buttons](#) to make objects on a layer visible in the 3D window.
- Selected - Lists only the object(s) currently selected in the 3D window. You can select multiple objects by ⇧ Shift RMB -clicking.
- Active - Lists only the active (often last selected) object.
- Same Types - Lists only those objects in the current scene that are of the same types as those selected in the 3d window.
- Groups - Lists only [Groups](#) and their members.
- Libraries - TODO
- Sequence - TODO
- Data Blocks - TODO
- User Preferences - TODO
- Key Maps - TODO

Beispiel



Das Outliner Fenster in der Listenansicht.

The outline example shows that the .blend file has three scenes: “Ratchet in Middle”, “Ratchet on Outside”, and “Ratchet Out White”. By clicking on the little plus-sign to the left of the name, the outline is expanded one level. This was done for the “Ratchet in Middle” scene. As you can see, this scene has some “World” material settings, a “Camera”, an “Empty”, a “HandleFixed” object... All objects that were added to the scene.

By clicking the plus-sign next to “ratchetgear”, we can see that it has some motion described by the “Animation” entry; that it was based on a “Circle” mesh, and that it is the parent of “HandleFixed.002”, which is in turn the parent of “Plane.003”, and so on.

The neat thing is: if you select any of these datablocks here, they will be selected in the 3D window as well, as far as this is possible. Pressing . NumPad with your mouse cursor in any 3D Window will center and align the view to that object. Very handy. Also, pressing X will delete it, as well as all the other hotkeys that operate on the currently selected object.

Einleitung

Wenn Sie Blender verwenden, erstellen Sie eine Welt in vier Dimensionen:

1. Links-Rechts, oder auch "x" Achse genannt.
2. Vorne-Hinten, oder auch "y" Achse genannt.
3. Oben-Unten, oder auch "z" Achse genannt.
4. Zeitbasiert, durch animierte Objekte, Materialien und Bewegungsabläufe in Frames.

Ein Problem besteht allerdings darin, dass Sie nur einen zweidimensionalen Bildschirm vor sich haben! Ihre Maus kann lediglich nach links-rechts und oben-unten bewegt werden. Sie können nicht in der Zeit zurückgehen und Sie können auch nicht in den Bildschirm greifen und ein Objekt bewegen.

Stattdessen müssen Sie Blender mitteilen, was er tun soll. Dieser Abschnitt erklärt Ihnen, wie Sie in der einzigartigen Blenderoberfläche in Ihrer virtuellen Welt navigieren.

Einleitung

Um in der Lage zu sein, im von Blender genutzten dreidimensionalen Raum zu arbeiten, muss man in der Lage sein, sowohl seinen Blickpunkt als auch die Blickrichtung innerhalb der Szene ändern zu können. Das hier erklärte 3D-Ansichtsfenster kann mit durchweg denselben Funktionalitäten wie die meisten anderen Fenster aufwarten. Beispielsweise ist es möglich, Felder mit Tasten und Panels zu verschieben und zu skalieren.



Mausknöpfe und Numpad

Bei Verwendung einer Maus mit weniger als drei Knöpfen oder einer Tastatur ohne Numpad bitte auf der Seite [Eingabegeräte](#) des Handbuchs nachschauen, wie in solchen Fällen in Blender zu verfahren ist.

Perspektivische und Orthographische Ansichten

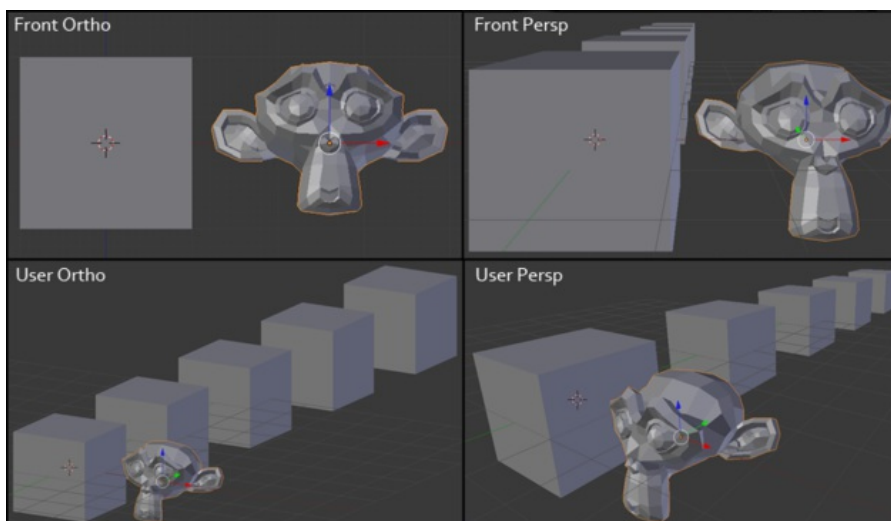
Modus: Alle Modi

Tastenkombination: 5 NumPad

Menü: Ansicht » Perspektive / Ansicht » Orthogonal

Beschreibung

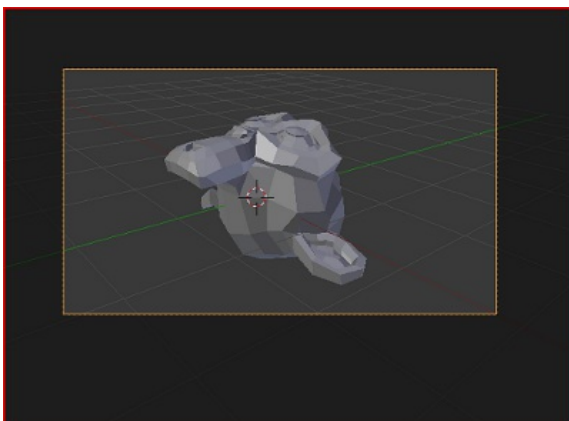
Jede 3D-Ansicht unterstützt zwei verschiedene Typen von Projektionen. Diese werden im unteren Bild *Orthographisch (links) und Perspektive (rechts) Projektionen* gezeigt.



Orthographische (links) und Perspektive (rechts) Projektionen.

Unser Auge ist an die perspektivische Ansicht gewöhnt, weil Objekte in der Ferne kleiner erscheinen. Die Orthographische Projektion erscheint oft auf den ersten Blick ein wenig seltsam, da sich Objekte unabhängig von ihrer Distanz nicht ihre Größe verändern. Sie erscheint wie die Betrachtung einer Szene mit einem unendlichen Fluchtpunkt. Gleichwohl ist die orthographische Ansicht sehr nützlich (es ist die Standardprojektion in Blender und vielen anderen 3D Programmen), da sie einen tieferen "technischen" Einblick in die Szene gewährt, die das Zeichnen und Beurteilen von Proportionen erleichtert.

Optionen



Demonstration der Kameraansicht.

Um die Projektionsart einer 3D-Ansicht zu wechseln, wählt man View » Orthographic oder die Menüoption View » Perspective. Mit dem Tastatur-Shortcut 5 NumPad kann man zwischen beiden Projektionsarten hin- und herschalten. Dabei hat die Änderung der 3D-Ansicht keinen Einfluss auf das Rendern, das standardmäßig perspektivisch erfolgt. Möchte man dennoch ein orthographisches

Rendern, wählt man die Kamera, öffnet das Object Data-Kontextmenü und drückt die Orthographic-Taste auf dem Lens-Panel.

Die Menü-Option View » Camera setzt die 3D-Ansicht auf die Kameraansicht (0 NumPad). Die Szene wird dann aus der Ansicht angezeigt, wie sie später gerendert wird (siehe *Demonstration der Kameraansicht*). Das gerenderte Bild wird alles enthalten, was sich innerhalb der gepunkteten orangefarbenen Linie befindet. Das Vergrößern und Verkleinern ist in dieser Ansicht möglich, aber um den Ansichtspunkt zu verändern, muss die Kamera bewegt oder rotiert werden.

Bei einer großen Szene zeigt die Kameraansicht möglicherweise nicht alle Objekte der Szene. Eine Ursache könnte die zu niedrig eingestellte [Clipping Distance](#) der Kamera sein. Die Kamera zeigt nur Objekte an, die sich innerhalb der Clipping-Reichweite befinden.

[Mehr zu Render Perspektiven »](#)

[Mehr zur Kameraansicht »](#)

[Mehr über Kamera-Clipping »](#)

Technische Details

Definition der Perspektive

Eine *perspektivische* Ansicht setzt sich geometrisch aus einer Szene in 3D und dem Platzieren eines Betrachters auf dem Fluchtpunkt o zusammen. Demgegenüber wird eine 2D-Perspektive auf einer Ebene (z.B. einem Blatt Papier) aufgebaut, für welche die 2D-Szene vor dem Fluchtpunkt o gezeichnet wird, und zwar im rechten Winkel zur Blickrichtung. Für jeden Punkt p in einer 3D-Szene wird eine Linie po gezeichnet, welche durch o und p verläuft. Der Schnittpunkt s zwischen dieser Linie po und der Ebene ist die perspektivische Projektion dieses Punktes. Projiziert man so alle Punkte p der Szene, erhält man eine perspektivische Darstellung.

Definition der Orthographie

In einer *orthographischen* Projektion gibt es zwar eine Blickrichtung, aber keinen Fluchtpunkt o . Die Linie ist in diesem Falle durch Punkt p gezeichnet, so dass sie parallel zur Blickrichtung liegt. Der Schnittpunkt s zwischen der Linie und der Ebene ist die orthographische Projektion des Punktes p . Projiziert man alle Punkte p der Szene auf diese Weise, erhält man eine orthographische Darstellung.

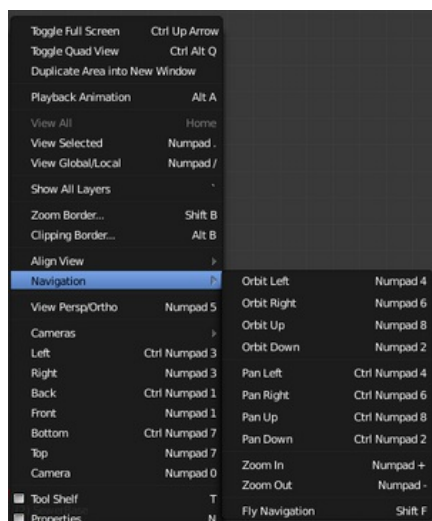
Rotieren der Ansicht

Modus: Alle Modi

Tastenkombination: MMB  / 2 NumPad / 4 NumPad / 6 NumPad / 8 NumPad / CtrlAlt Wheel 

Menü: View » Navigation

Beschreibung




A 3D viewport's View menu.

Blender provides four default viewing directions: Side, Front, Top and Camera view. Blender uses a right-angled “Cartesian” coordinate system with the Z axis pointing upwards. “Side” corresponds to looking along the X axis, in the negative direction, “Front” along the Y axis, and “top” along the Z axis. The Camera view shows the current scene as seen from the camera view point.

Optionen

You can select the viewing direction for a 3D viewport with the View menu entries, or by pressing the hotkeys 3 NumPad for “side”, 1 NumPad for “front”, 7 NumPad for “top”. You can select the opposite directions if you hold Ctrl while using the same numpad shortcuts. Finally 0 NumPad gives access to the “camera” viewpoint.

Apart from these four default directions, the view can be rotated to any angle you wish. Click and drag MMB  on the viewport's area.

If you start in the middle of the window and move up and down or left and right, the view is rotated around the middle of the window. Alternatively, if the Emulate 3 button mouse option is select in the User Preferences you can press and hold Alt while dragging LMB in the viewport's area.

To change the viewing angle in discrete steps, use 8 NumPad and 2 NumPad (which correspond to vertical MMB dragging, from any viewpoint), or use 4 NumPad and 6 NumPad (or CtrlAlt Wheel) to rotate the scene around the Z global axis from your current point of view.

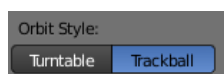
Hotkeys

Remember that most hotkeys affect **the active window** (the one that has focus), so check that the mouse cursor is in the area you want to work in before your use the hotkeys.

TrackBall/Turntable

By default, when you rotate the view as described above, you are rotating the scene as though you are rolling your hand across a **“trackball”**. For some users this is intuitive and for others it is not. If you feel you are having difficulties with this style of 3D window rotation you can switch to the **“turntable”** style.

The Turntable style is fashioned more like a record player where you have two axes of rotation available, and the world seems to have a better definition of what is “Up” and “Down” in it. The downside to using the Turntable style is that you lose some flexibility when working with your objects. However, you gain the sense of “Up” and “Down” which can help if you are feeling disoriented. Of course you can always switch between the styles depending on what you are working on.



View rotation.

To change the rotation “style”, use the [User Preferences window](#). Click on the Input button and you will see an option for choosing the Orbit style. There are two additional checkboxes for controlling the display in the 3D window in the Interface tab in the User Preferences. Auto Perspective will automatically switch to perspective whenever the view is rotated using MMB. Rotate Around Selection will rotate the view around the center of the current selection. If there is no selection at that moment (e.g. if you used A to deselect everything), the last selection will be used anyway.

Schwenken der Ansicht

Modus: Alle Modi

Tastenkombination: ⇧ Shift MMB / Ctrl2 NumPad / Ctrl4 NumPad / Ctrl6 NumPad / Ctrl8 NumPad / ⇧ ShiftAlt LMB

Menü: View → Navigation

Beschreibung

Zum Schwenken der Ansicht, halten Sie ⇧ Shift und ziehen MMB in der 3D Ansicht. For discrete steps, use the hotkeys Ctrl8 NumPad, Ctrl2 NumPad, Ctrl4 NumPad and Ctrl6 NumPad as with rotating (note: you can replace Ctrl by ⇧ Shift). For those without a middle mouse button, you can hold ⇧ Shift Alt while dragging with LMB.

Vergrößern der Ansicht

Mode: All modes

Hotkey: Ctrl MMB / Wheel / + NumPad / - NumPad

Menu: View → Navigation

Beschreibung

You can zoom in and out by holding down Ctrl and dragging MMB. The hotkeys are + NumPad and - NumPad. The View » Navigation sub-menu holds these functions too as well. Refer to the 3D viewport's View menu image above for more information.

If you have a wheel mouse, you can perform all of the actions in the 3D viewport that you would do with + NumPad and - NumPad by rotating the Wheel. To zoom a Buttons window, hold Ctrl MMB and move your mouse up and down.

If You Get Lost...

If you get lost in 3D space, which is not uncommon, two hotkeys will help you: ↶ Home changes the view so that you can see all objects (View » View All menu entry), while . NumPad zooms the view to the currently selected objects when in perspective mode (View » View Selected menu entry).

Rahmen vergrößern

Das Werkzeug Rahmen zoomen... erlaubt Ihnen einen rechteckigen Bereich festzulegen, und diesen Bereich zu vergrößern damit diese die 3D Ansicht füllt.

Sie können dieses durch das Menü Ansicht erreichen, oder die Tastenkombination ⇧ ShiftB dann klicken und ziehen Sie einen

rechteckigen Bereich um darein zu vergrößern.

Ausrichten der Ansicht

Ansicht ausrichten

Diese Optionen erlauben Ihnen die Ansicht auf verschiedene Weisen auszurichten und zu orientieren. Sie finden diese in dem Ansichtsmenü

Ansicht an Ausgewählten ausrichten

These options align your view with specified local axes of the selected object or, in Edit mode, with the normal of the selected face.

Top ⇧ Shift7 NumPad
 Bottom ⇧ ShiftCtrl7 NumPad
 Front ⇧ Shift1 NumPad
 Back ⇧ ShiftCtrl1 NumPad
 Right ⇧ Shift3 NumPad
 Left ⇧ ShiftCtrl3 NumPad

Cursor zentrieren und alle zeigen (⇧ ShiftC)

moves the cursor back to the origin **and** zooms in/out so that you can see everything in your scene.

Aktive Kamera an Ansicht ausrichten, CtrlAlt0 NumPad

Gives your active camera the current viewpoint

Ausgewählte zeigen, . NumPad

Focuses view on currently selected object/s by centering them in the viewport, and zooming in until they fill the screen.

Ansicht auf Cursor zentrieren, Ctrl. NumPad

Centers view to 3D-cursor

Ausgewählte zeigen

See above

Alles zeigen ↵ Home

Frames all the objects in the scene, so they are visible in the viewport.

Lokale und Globale Ansicht

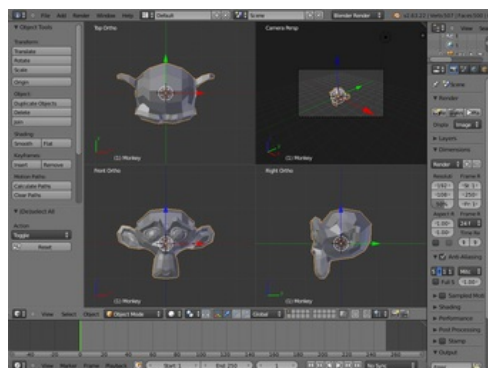
You can toggle between Local and Global view by selecting the option from the View Menu or using the shortcut / NumPad. Local view isolates the selected object or objects, so that they are the only ones visible in the viewport. This is useful for working on objects that are obscured by other ones, or have heavy geometry. Press / NumPad to return to Global View

Vierer-Ansicht

Modus: Alle Modi

Tastenkombination: CtrlAltQ

Menü: View » Vierer-Ansicht an/aus



Quad View

Das Aktivieren der Vierer-Ansicht wird das 3D Fenster in vier Ansichten aufteilen : Top Ortho, Front Ortho, Right Ortho and Camera Perspective. This view will allow you to instantly see your model from a number of view points. In this arrangement, you can zoom and pan each view independantly but you cannot rotate the view. Note that this is different from splitting the windows and aligning the view manually. In Quad View, the four views are still part of a single 3D window. If you want to be able to rotate each view, you will need to

split the 3D window into separate windows.

[Read more about splitting windows »](#)

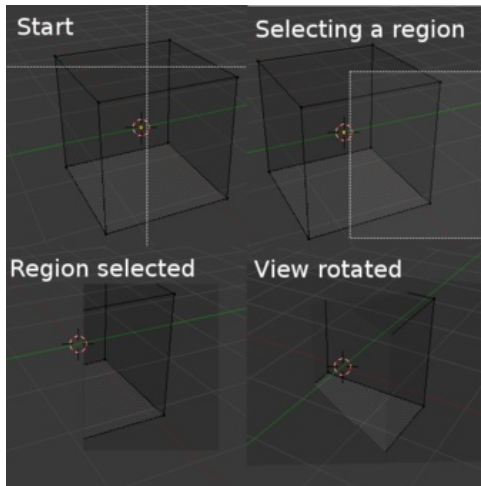
Clipping-Rahmen ansehen

Modus: Alle Modi

Tastenkombination: AltB

Menü: View » Ansichts-Clipping-Rahmen setzen

Beschreibung



Region/Volume clipping.

To assist in the process of working with complex models and scenes, you can change the view clipping to visually isolate what you're working on.



Das Werkzeug wird nur anzeigen was sich in einem von Ihnen definierten Volumen befindet.

Once activated with AltB, you have to draw a rectangle with the mouse, in the wanted 3D view. The created clipping volume will then be:

- A right-angled [parallelepiped](#) (of infinite length) if your view is orthographic.
- A rectangular-based pyramid (of infinite height) if your view is in perspective.

Um dieses Clipping zu löschen, drücken Sie AltB erneut.

Beispiel

The *Region/Volume clipping* image shows an example of using the clipping tool with a cube. Start by activating the tool with AltB (upper left of the image). This will generate a dashed cross-hair cursor. Click with the LMB  and drag out a rectangular region shown in the upper right. Now a region is defined and clipping is applied against that region in 3D space. Notice that part of the cube is now invisible or clipped. Use the MMB  to rotate the view and you will see that only what is inside the pyramidal volume is visible. All the editing tools still function as normal but only within the pyramidal clipping volume.

The dark gray area is the clipping volume itself. Once clipping is deactivated with another AltB, all of 3D space will become visible again.

Einleitung

Um in Blenders drei dmensionalen Raum arbeiten zu können müssen alle Sichtpunkte und -richtungen erreichbar sein. Auch wenn hier die 3D Ansicht im Detail beschrieben wird, so benutzen viele andere Fenster von Blender ähnliche Operationen zum Navigieren und Manipulieren von Objekten. Zum Beispiel funktionieren Zoom, Auswahl, Bewegen und das Löschen von verschiedenen Objekten mit den gleich Tastenkombinationen und Mausgesten.



Maus Tasten und der Numblock

Falls nur eine Maus mit weniger als drei Tasten und/oder eine Tastatur ohne Numblock vorliegen, siehe die [Tastatur und Maus](#) Seite des Handbuchs um zu lernen wie man sie in Blender benutzt.

perspektivische und orthografische Ansicht

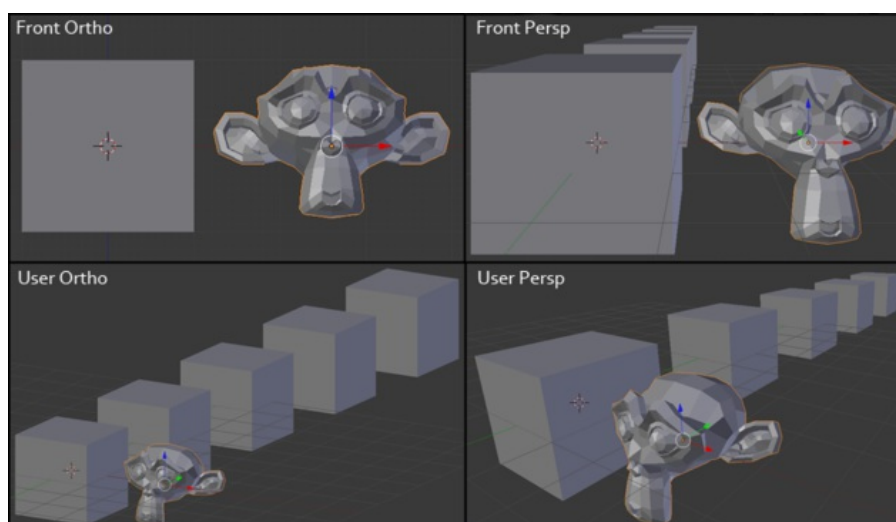
Mode: Alle Modi

Hotkey: 5 NumPad

Menu: Ansicht » perspektivisch / Ansicht » orthografisch

Beschreibung

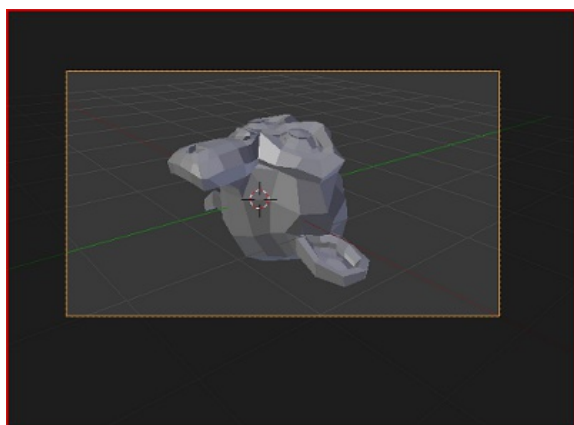
Jedes 3D Ansichtsfenster hat zwei verschiedene Darstellungstypen. Diese sind vorgeführt in den *ortografischen (links)* und der *perspektivischen (rechts)* Projektions Bildern unten.



ortografischen (links) und der perspektivischen (rechts) Projektionen

Unser Auge ist an perspektivisches Sehen gewöhnt: weit entfernte Objekte erscheinen kleiner. Orthografische Projektionen kommen einem erst seltsam vor, da Objekte immer die gleiche Größe behalten, unabhängig von ihrer Entfernung von der Kamera. Trotzdem ist es sehr nützlich, da es einen "technischen" Einblick in die Szene gibt, der es einfacher macht bestimmte Proportionen abzuschätzen und zu zeichnen.

Optionen



Demonstration der Kamera Sicht

Um die Darstellungsart der 3D Sicht zu verändern wähle entweder Ansicht » Orthographisch oder Ansicht » Perspektivisch aus den Menü Einträgen. Die Taste 5 NumPad wechselt zwischen den beiden. Die 3D Ansicht zu verändern, verändert aber nicht wie die Szene gerendert wird. Das passiert standardmäßig immer als perspektivische Projektion. Wenn es nötig sein sollte orthografisch zu rendern, muss die Kamera ausgewählt und im Objekt Daten Kontextmenü der orthografisch Knopf im Linsen Abschnitt gedrückt

werden.

Der Ansicht » Kamera Eintrag stellt die 3D Ansicht auf den Kamera Modus ein (Ein und Aus auch auf0 NumPad). Die Szene wird dann aus der Kameraperspektive der aktiven Kamera dargestellt, also so wie die Szene später gerendert werden soll. Auf dem gerenderten Bild wird alles innerhalb der gestrichelten Orange farbigen Linie zu sehen sein. Herein und heraus zoomen ist zwar möglich, ändert aber nicht das Renderergebnis - dafür muss die Kamera bewegt oder vergrößert werden.

Falls die Szene zu groß ist kann es sein das nicht alle Objekte in der Szene dargestellt werden können. Es kann sein das die ["clipping" Entfernung](#) zu niedrig ist - sie steht für die Sichtweite der Kamera.

[Lerne hier mehr über Render Perspektiven »](#)

[Lerne hier mehr über Kameraperspektive »](#)

[Lerne hier mehr über Kamera clipping »](#)

Technische Details

Perspektivische Projektion



Eine *perspektivische Sicht* wird aus einer Szene in 3D und einem Beobachter bei Punkt o geometrisch konstruiert. Die zu erzeugende perspektivische 2D Szene wird erstellt indem man eine Ebene auf die die 2D Szene projiziert wird, senkrecht zur Sichtrichtung vor den Punkt o gesetzt. Für jeden Punkt p in der 3D Szene wird eine Linie po von o nach p gezogen. Wo diese Linie die Ebene schneidet wird der Punkt aus der 3D Szene hinprojiziert. Durch Projizieren aller Punkte der Szene auf die 2D Ebene entsteht eine perspektivische Ansicht der 3D Szene.

Orthografische Projektion

Eine *orthografische Sicht* hat zwar eine Blickrichtung aber keinen Beobachtungspunkt. In diesem Fall wird für jeden Punkt in der Szene eine Linie Parallel zur Sichtrichtung gezogen, die auf der Ebene ggf. einen entsprechenden Punkt erzeugt.

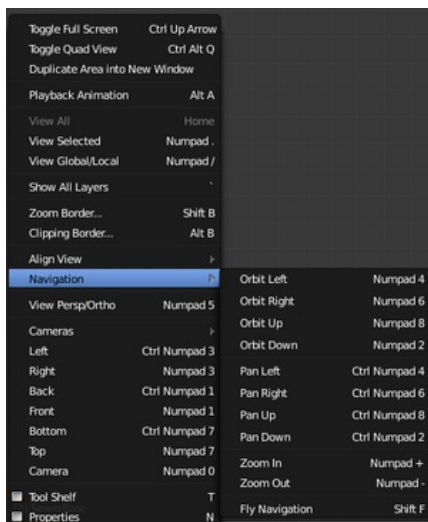
Die Blickrichtung drehen

Mode: Alle Modi

Hotkey: MMB  / 2 NumPad / 4 NumPad / 6 NumPad / 8 NumPad / CtrlAlt Wheel 

Menu: Ansicht » Navigation


Beschreibung



A 3D viewport's Ansicht Menü

Blender hat vier standard Blickrichtungen: von der Seite, von Vorne, von Oben und aus der Sicht der Kamera. Blender benutzt ein rechtwinkliges kartesisches Koordinatensystem mit einer nach oben gerichteten z-Achse. Die Seitenansicht ist die Blickrichtung in negative Richtung auf der x-Achse. Die Ansicht von Vorne ist die Blickrichtung in negative Richtung auf der y-Achse. Die Ansicht von Vorne ist die Blickrichtung in negative Richtung auf der z-Achse. Die Kamerasicht zeigt die Szene aus der Sicht der Kamera.

Optionen



Die Blickrichtungen kann um jeden gewünschten Winkel gedreht werden. Klicken, halten von MMB  und bewegen der Maus innerhalb der 3D Ansicht dreht die Kamera. Alternativ kann der dritte Mausbutton emuliert werden wenn emuliere dritten Mausbutton in den Benutzereinstellungen ausgewählt ist. Dann wird die Ansicht durch halten der Alt Taste und bewegen der Maus gedreht.

Für mehr Kontrolle und diskrete Schritte, deren Größe (standardmäßig 10°) in den Benutzereinstellungen eingestellt werden kann, können 8 NumPad für Drehung nach oben, 2 NumPad für Drehung nach unten, 4 NumPad für Drehung nach links und 6 NumPad für Drehung nach rechts gedrückt werden.

Tasten für des Drehen der Ansicht sind:

- 8 NumPad für Drehung nach oben
- 4 NumPad für Drehung nach links
- 6 NumPad für Drehung nach rechts
- 2 NumPad für Drehung nach unten

Die gleichen Operationen können auch mit dem Mausrad und einer Kontrolltaste ausgeführt werden ausgeführt werden:

- Wheel  CtrlAlt dreht die Ansicht um einen Fokuspunkt um die Horizontale.
- Wheel  ShiftAlt dreht die Ansicht um einen Fokuspunkt um die Vertikale.

Alle Drehungen beziehen sich auf das globale Koordinatensystem.

Beachte

Schnellasten beeinflussen immer das aktive Fenster, also das in dem sich gerade der Mauscursor befindet.

TrackBall/Turntable

Standardmäßig ist der Modus des oben beschriebenen Rotierens auf "**trackball**" eingestellt. Das Objekt verhält sich dann als ob die Szene ein Kugel ist die gerollt wird. Wenn mit diesem Modus Verwirrungen aufkommen kann auf den "**turntable**" Rotationstil gewechselt werden.

Der Turntable verhält sich wie eine Schallplatte die man um die Rotations Achse der Platte und um die Achse zwischen den zwei Randpunkten, an denen man die Platte festhält, drehen kann. Es geht allerdings Flexibilität verloren im vergleich zum Trackball.



Ansicht Rotation

Der Modus kann in den [Benutzer Einstellungen](#) gewechselt werden. Im Input Reiter auf der linken Seite befindet sich die Orbit style Einstellung. Im Interface Reiter gibt es zwei weitere Einstellungen zur Rotation: die Auto Perspective Checkbox wird die Ansicht automatisch auf perspektivisch umstellen wenn gedreht wird und Rotate Around Selection macht das Objekt Zentrum des aktiven Objekts zum Drehpunkt. Wenn grad kein Objekt ausgewählt ist wird das Zentrum des zuletzt ausgewählten Objekts genutzt.

Die Ansicht verschieben

Mode: Alle Modi

Hotkey:  Shift MMB  / Ctrl2 NumPad / Ctrl4 NumPad / Ctrl6 NumPad / Ctrl8 NumPad /  ShiftAlt LMB 



Menu: Ansicht → Navigation

Beschreibung

Um die Ansicht mit der Mausbewegung zu verschieben:

-  Shift MMB 

Um die Ansicht mit dem Mausrad zu verschieben:

- Wheel  Ctrl bewegt die Ansicht horizontal, parallel zur Sichtebene.
- Wheel  Shift bewegt die Ansicht vertikal, parallel zur Sichtebene.

Um die Ansicht in Diskreten Schritten zu verschieben:

- Ctrl2 NumPad bewegt die Ansicht nach unten.
- Ctrl4 NumPad bewegt die Ansicht nach links.
- Ctrl6 NumPad bewegt die Ansicht nach rechts.
- Ctrl8 NumPad bewegt die Ansicht nach oben.

Die Ansicht Zoomen


Mode: Alle Modi


Hotkey: Ctrl MMB  / Wheel  / + NumPad / - NumPad

Menu: Ansicht → Navigation

Beschreibung


Es kann gezoomt werden durch:

- Wheel  normales zoomen
- + NumPad hereinzoomen


- NumPad herauszoomen
- Ctrl MMB  und bewegen der Maus - normales zoomen

Das Ansicht » Navigation Sub-Menü enthält ebenfalls diese Operationen.

Im Falle des Orientierungsverlusts...

Falls die Orientierung im 3D Raum verloren geht gibt es die  Home/Ansicht » Sieh Alle Funktion die die Ansicht so verändert das alle Objekte in der Szene zu sehen sind. Die Funktion View » View Selected/. NumPad zoomt die Ansicht zum momentan ausgewählten Objekt.

Auswahlzoom

Das Auswahlzoom Werkzeug erlaubt dir eine rechteckige Region auszuwählen und auf sie zu zoomen, sodass sie die 3D Ansicht ausfüllt. Diese Funktion ist durch das Ansicht Menü verfügbar ebenso wie durch die Tastenkombination  ShiftB. Markiere das Rechteck durch klicken und aufziehen des Rechtecks.

Die Ansicht ausrichten

Diese Optionen ermöglicht es die Sicht nach den Achsen des Koordinatensystems auszurichten. Sie befinden sich ebenso im Ansicht Menü.

Die Standardansichten zusammengefasst:

- 1 NumPad für die Ansicht von Vorne.
- 1 NumPadCtrl für die Ansicht von Hinten.
- 3 NumPad für die Ansicht von Links.
- 3 NumPadCtrl für die Ansicht von Rechts.
- 7 NumPad für die Ansicht von Oben.
- 7 NumPadCtrl für die Ansicht von Unten.
- 0 NumPad für die Kameraperspektive
- CtrlAlt0 NumPad Verschiebt die aktive Kamera auf die Sichtausrichtung.

Lokale und Globale Ansicht

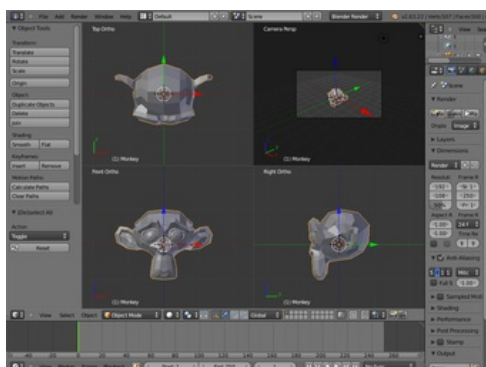
Es kann zwischen lokaler und globaler Ansicht mit / NumPad und der entsprechenden Funktion im Ansicht Menü gewechselt werden. Diese blendet alle anderen Objekte bis auf das Ausgewählte aus.

Quad Ansicht

Mode: Alle Modi

Hotkey: CtrlAltQ

Menu: Ansicht » Toggle Quad Ansicht



Quad Ansicht

Quad Ansicht einzuschalten teilt das 3D Fenster in vier Ansichten: von oben, vorne und rechts orthografisch und die Kameraperspektive. Das erlaubt zusätzliche Übersicht über mehrere Ansichten eines Objekts gleichzeitig. Jedes Fenster kann unabhängig von den anderen gezoomt und bewegt werden, nur eine Rotation der Sicht ist nicht möglich. Diese Ansicht ist nicht gleich dem teilen der Fenster in vier unabhängige 3D Ansichten, denn alle vier Ansichten sind bei der Quad Ansicht Teil derselben Ansicht.

[Lerne hier mehr über das teilen von Fenstern. »](#)

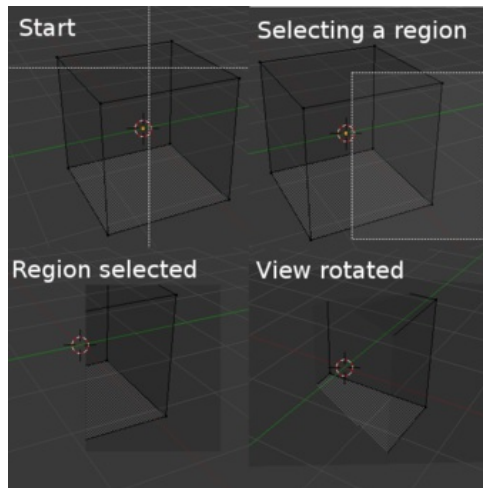
Schneide Grenze anzeigen

Mode: Alle Modi

Hotkey: AltB

Menu: Ansicht » Setze Schneide Grenze

Beschreibung



Regionen/Volumen schneiden

Um bei komplexen modellen und Szene die Übersicht zu behalten kann das Ansichts clipping eingestellt werden um visuell zu isolieren woran gearbeitet wird.

Dieses Werkzeug zeigt nur an was innerhalb des definierten Volumens liegt.

Es wird aktiviert mit AltB. Nun muss in der 3D Ansicht ein Rechteck gezogen werden. Alles innerhalb des Rechtecks wird sichtbar bleiben. Das ergebnis ist entweder


- rechtwinkliges Trapez mit unendlicher Höhe wenn der Befehl in orthografischer Perspektive ausgeführt wurde,

oder

- eine Pyramide mit rechtwinkliger Grundfläche unendlicher Höhe dessen Seitenflächen die Kanten des gezogenen Rechtecks beinhalten.

Um die Auswahl aufzuheben drücke nochmal AltB.

Beispiel

Das *Regionen/Volumen schneiden* Bild zeigt ein Beispiel des Schneide Werkzeug an einem Kubus. Das einschalten des Tools mit AltB erzeugt den Fadenkreuz Mauszeiger. LMB  und das ziehen des Mauszeigers erzeugt die rechtwinklige Region oben rechts im Bild. Die Region ist nun definiert und das schneiden wird nun auf diese Region des 3D Raums angewandt. Zu beachten ist das ein Teil des Würfels nun nicht mehr sichtbar und weggeschnitten ist. Durch verändern der Ansicht wird klar das nur das innere des definierten Raums sichtbar ist. Alle Bearbeitungsfunktionen funktionieren normal, aber nur innerhalb des Raums. Die dunklere Fläche ist das Volumen das beim Schnitt übrig bleibt.

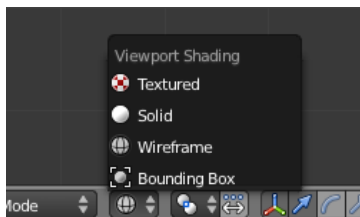
Darstellungsmodi

Mode: All modes

Hotkey: Z / ⇧ ShiftZ / AltZ / ⇧ ShiftAltZ / D

Beschreibung

Abhängig von der Leistung des Computers und der anliegenden Aufgaben ist es sinnvoll zwischen bestimmten Darstellungsmodi zu wechseln.



Ein 3D Darstellungsmodibutton

Texturiert

Stellt Objekte mit UV texturen und OpenGL Beleuchtung dar.

Shaded

Nähert Texturen und Beleuchtung für jeden Vertex und stellt Verläufe zwischen den Vertices dar. Es ist zwar nicht so genau wie die Render Engine aber dafür sehr viel schneller.

Solid

Ist der Standarddarstellungsmodus für Objekte. Oberflächen werden mit soliden Farben und OpenGL Beleuchtung dargestellt. Der Darstellungsmodus ist nicht von Beleuchtung abhängig und kann im Benutzereinstellungen>>System & OpenGL>>Solid OpenGL Lights eingestellt werden.

[Lies hier mehr über Systemeinstellungen.](#)

Drahtgestell

Stellt nur die Kanten von Objekten dar.

Bounding Box

Objekte werden durch Boxen mit den Hauptdimensionen der Objekte ersetzt.

Es kann zwischen den Darstellungsmodi gewechselt werden durch:

- das benutzen des Darstellungsmodusmenüs im 3D Ansichtsbalken.
- das Drücken von D um das Darstellungsmodusmenü aufzurufen
- das benutzen der Z-basierten Tastenkombinationen:

Darstellungsmodi und Z-basierte Tastenkombinationen

Z	wechselt zwischen den Drahtgestell und Solid Darstellungsmodi.
⇧ ShiftZ	wechselt zwischen den Drahtgestell und Shaded Darstellungsmodi.
AltZ	wechselt zwischen Solid und Texturiert.
⇧ ShiftAltZ	wechselt zum Texturierten Darstellungsmodus.

Ansichtseigenschaften Panel

Mode: Alle Modi

Panel: Ansichtseigenschaften

Menu: Ansicht » Ansichtseigenschaften...

Beschreibung

Zusätzlich zu den oben beschriebenen Kontrollen gibt das [Template:Ansichtseigenschaften](#) Menü einige zusätzliche Optionen zur 3D Ansicht.

Ansicht

Linse

Control the focal length of the 3d view camera in millimeters, unlike a [rendering camera](#)

Lock to Object

Durch eingeben eines Objektnamens in das Objekt Feld, wird die Ansicht auf das Objekt festgelegt. Es wird in der Ansicht Zentriert, mit Ausnahme der aktiven Kamera.

Falls eine Armatur zentriert wird, kann die Ansicht weiter auf einen Bone Zentriert werden durch eingeben des Names.

Lock to Cursor

zentriert die Ansicht auf den 3D Cursor.

Lock Camera to View

Wenn die Kamerasicht aktiviert ist kann diese Option dazu benutzt werden die Kamera zu bewegen ohne das die Kamerasicht dafür verlassen werden müsste.

Clip Start und Clip End

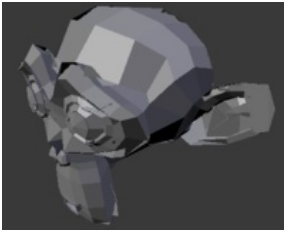
Stelle die minimale und die maximale Entfernung in der Ansicht ein, ab der Objekte dargestellt werden.

Notiz

Eine große Clipping Entfernung erlaubt zwar das ansehen von nahen und entfernten Objekten, reduziert aber die Tiefengenaugigkeit.



Form ohne Clipping-Artefakten



Form mit Clipping-Artefakten

Um das zu verhindern:

- erhöhe das Near Clipping beim arbeiten an großen Szenen.
- verringere das Far Clipping bei nahen arbeiten an Objekten.

Wenn die orthografische Ansicht eingeschaltet ist, wird die entfernte Clipping Entfernung genommen. Sehr hohe Werte können trotzdem Artefakte verursachen.

Das ist nicht ein blenderspezifisches Problem, da alle OpenGL/DirectX Grafikanwendungen diese Grenzen haben.

Lockale kamera

Die in dieser Ansicht aktive Kamera

3D Cursor Position

Hier kann die Position des 3D Cursors exakt eingegeben werden.

Item

Dieser Bereich zeigt das aktuell ausgewählte Objekt.

Darstellung**Nur Render**

Stellt nur Objekte dar die gerendert werden.

Displays only items that will be rendered.

Markierung der Außenkanten

Schaltet die orange Markierung der Außenkanten der ausgewählten Objekte ein oder aus.

Alle Objekt Zentren

Schaltet ein das alle Objektzentren unabhängig von der Auswahl dargestellt werden

Beziehungslinien

Kontrolliert ob die gestrichelten Parent, Beschränkungs, Haken, etc. Linien dargestellt werden.

Alle Kanten

Wenn das Drahtoverlay im Objektkontext aktiviert ist, werden alle Kanten von Drahtgestellen dargestellt.

Gitter Boden

Schaltet den Gitter Boden in nicht orthografischen Ansichten aus.

X Axis, Y Axis, Z Axis

Schaltet die dargestllten Achsen ein oder aus.

Linien

Stellt die Anzahl von Linien im Gitterboden ein.

Skalierung

Kontrolliere die Ausmaße des des Gitterbodens hier.

Unterteilung

Stelle hier die Unterteilung beim herinzoomen in jeder Zelle des Gitterbodens hier ein.

Shading

Stellt ein wie Objekte in der 3D Ansicht dargestellt werden.

Textured Solid

Zeigt für Flächen eigenstellte Texturen dar.

Toggle Quad View

Stellt die Quad Unterteilung ein oder aus. [Lies hier mehr über die Qaudunterteilung.](#)

Hintergrund Bild

Mode: All Modi

Panel: Hintergrundbild

Menu: Ansicht » Ansichtseigenschaften

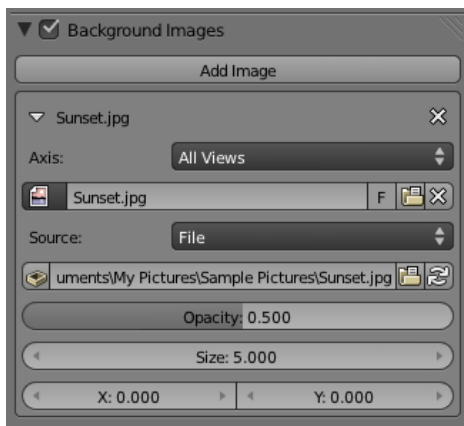
Ein Hintergrund Bild oder Film kann in vielen Fällen sehr nützlich sein, zum Beispiel beim modellieren, anmalen oder animieren.



Es gibt eine wichtige Dinge die bei Hintergrundbildern zu beachten sind:

- Sie sind speziell für ein Fenster, es ist also möglich verschiedene Hintergrundbilder für verschiedene Fenster einzustellen.
- *Sie sind nur in den direkten orthografischen Ansicht von den Seiten verfügbar.* Das Bild in einem Fenster bleibt dasselbe.
- Die Größe hängt vom Vergrößerungsfaktor des Fensters ab.

Einstellungen



Das Hintergrund Bild Panel

Blender verwaltet diese Option mit dem Hintergrundbildmenü in den Ansichtseigenschaften. Die Achse gibt an in welche Ansicht das Bild geladen wird.

Wenn ein Bild geladen ist stehen folgende Optionen zur Verfügung:

Quelle

Stellt ein welches Bild benutzt wird.

File

Benutzt ein Bild.

Bilddatei

Repräsentiert die Bilddatei die im Datenblock verlinkt ist.

Sequenz

Benutzt eine Sequenz von nummerierten Bildern.

Frames

Stellt die Nummer der Bilder in der Sequenz ein.

Start

Stellt den Start-Frame in der Sequenz ein.

Offset

Stellt einen Offset bei der Bildsequenz ein.

Felder

Stellt die Anzahl der Felder pro eingestelltem Frame ein.

Auto Refresh

erneuert das Bild bei Framwechseln
Always refresh the image on frame changes

Cyclic

Wiederhole die Bilderfrequenz

Film

Benutzt einen Film:

Stelle Framezahl auf Filmlänge

Stellt die Framezahl der Szene auf die Länge des verwendeten Films passend ein.

Erzeugt

Benutze ein in Blender erzeugtes Bild

Breite, Höhe

Stellt Höhe und Breite des verwendeten Bilds ein.

Blank

Erzeugt ein leeres Bild

UV Gitter

Erzeugt ein UV Gitter für Testzwecke ein.

Farbgitter

Erzeugt ein farbiges UV Gitter für Testzwecke ein.

Durchsichtigkeit

Dieser Regler stellt die Durchsichtigkeit des Hintergrundbilds ein **0.0** – vollständig sichtbar – **1.0** – vollständig durchsichtig.

Größe

Stellt Größe und Skalierung für das Hintergrundbild in Blendereinheiten ein.

X Verschiebung, Y Verschiebung

Stellt eine horizontale und vertikale Verschiebung für das Hintergrundbild ein. Standardmäßig ist es im Fenster zentriert.



Benutze eine Proxy mit kleiner Auflösung

Die die Leistung des PCs effektiv zu nutzen können Proxies mit kleinerer Auflösung benutzt werden. Bei einer Monitorauflösung von 800x600 muss das Hintergrundbild auch nicht größer sein. Falls das Vergleichsbild 2048x2048 groß ist verschwendet der PC nur Leistung. Versuch stattdessen es mit einem Grafikprogramm herunterzuskalieren.

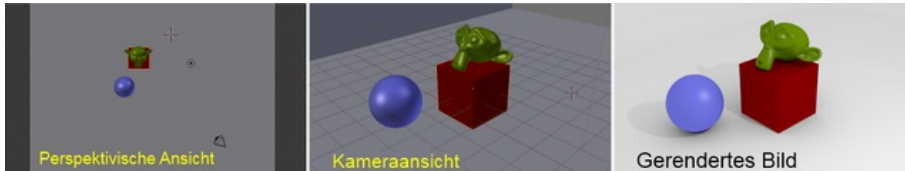
Kameraansicht

Modus: Alle Modi

Tastenkombination: 0 NumPad

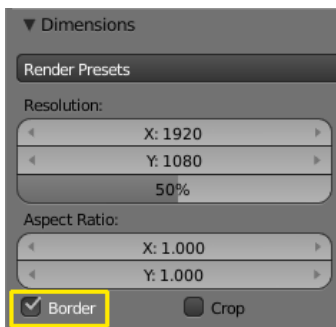
Menü: Ansicht » Kamera » Aktive Kamera

Die Kameraansicht kann dazu benutzt werden virtuell Bilder zu komponieren und eine Vorschau der gerenderten Szene zu bekommen. Durch drücken von 0 NumPad wird die Szene aus Sicht der momentan aktiven Kamera angezeigt. In dieser Ansicht können Sie auch den Render Rand setzen, welcher die Teile der zu rendernden Kameraansicht definiert.



Die Kameraansicht stellt eine Vorschau des endgültig gerenderten Bildes bereit.

Render Rand



Render Border toggle

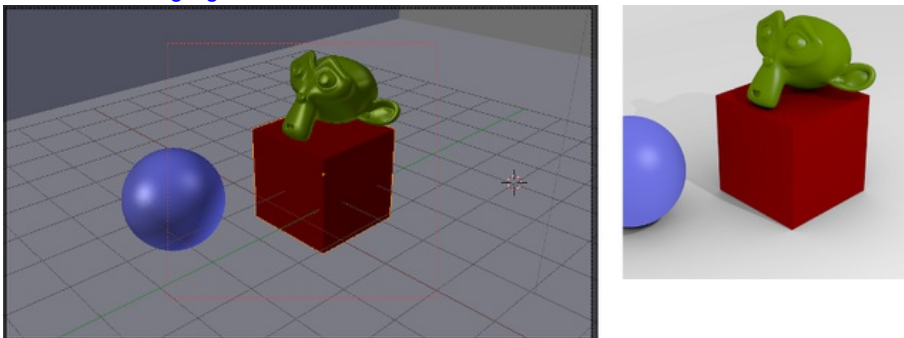
While in camera view, you can define a Render Border by pressing \diamond ShiftB. This will allow you to draw out a dotted orange rectangle within the camera view. Your renders will now be limited to the part of scene visible within the render border. This can be very useful for reducing render times. The border can be disabled by disabling the Border option in the Dimensions panel of the Render context or by using \diamond ShiftB to set a Render Border larger than the camera view.

Anti-Aliasing und Optionen zum Weichzeichnen mit Rändern

Beachten Sie dass wenn Render Ränder aktiviert sind, wird die Vollabtastung beim Anti-Aliasing deaktiviert, während die Abtastbewegungsunschärfe verfügbar wird.

[Mehr über Anti-Aliasing »](#)

[Mehr über Bewegungsunschärfe »](#)



Render border and associated render.

[Mehr über Render Möglichkeiten zur Ausgabe »](#)

[Mehr über Kameras »](#)

Layers

Mode: Object modus

Panel: Relations (Object context)

Hotkey: M

Menu: Object » Move to Layer...

3D Szenen werden meistens exponentiell verwirrender bei steigender Komplexität. Der Künstler braucht aber häufig individuelle präzise Kontrolle darüber wie Objekte ausgeleuchtet werden ohne bei der Bearbeitung durch andere Objekte gestört zu werden. Zu diesem und anderen Zwecken können Objekte in "Ebenen" strukturiert werden. Das ermöglicht es folgendes zu tun:

- selectiv Objekte auf bestimmten Ebenen darzustellen, durch auswählen der entsprechenden Ebenen im 3D Ansicht Menübalken.
- [Beleuchtung](#) zu kontrollieren die, durch begrenzen von Lampen auf bestimmte Ebenen.
- zu kontrollieren welche Kräfte [Partikel Systeme](#) beeinflussen.
- zu kontrollieren welche Ebenen gerendert werden (und damit welche Objekte) und welche Eigenschaften und Kanäle zum Zusammenfügen verfügbar gemacht werden durch Benutzung von [render layers](#)

Armaturen können sehr komplex werden mit verschiedenen Kochentypen, Reglern, Lösem, benutzerdefinierten Form, etc.. Außerdem sind sie natürlicherweise so dicht das man schnell den Überblick verliert. Aus diesem Grund hat Blender Ebenen die nur für Armaturen reserviert sind. Armatur Ebenen sind Objekt Ebenen sehr ähnlich, es können ebenso Armaturen auf verschiedene Ebenen verteilt und nur die angezeigt werden an denen gearbeitet wird.

[Lerne mehr über Armatur Ebenen »](#)



Arbeiten mit Ebenen

3D Ebenen sind insofern anders als Ebenen in 2D Grafiksoftware das ihre Reihenfolge keinen Einfluss auf das Ergebnis hat und hauptsächlich dazu da sind die Szene zu organisieren.

Es werden nur die zum Rendern ausgewählten Ebenen gerendert. Falls alle Lichtquelle auf einer Ebene sind die *nicht ausgewählt* ist, wird nichts im Ergebnis zu sehen sein.

[Gruppen und Parent-Hierarchie](#) sind andere Möglichkeiten um Objekte logisch zu ordnen.

Ebenen ansehen

Blender hat zwanzig Ebenen deren Sichtbarkeit mit den kleinen unmarkierten buttons (siehe 3D Ansicht Ebenen Buttons) ein- und ausgeschaltet werden können. Um einen Ebene auszuwählen klicke auf den entsprechenden Button mit LMB  oder ⇧ Shift LMB  um mehrere Ebenen an- oder abzuwählen.



3D Ansicht Ebenen Buttons

Um die Ebenen 1 bis 10 auszuwählen können alternativ auch entsprechend die Tasten von 1 bis 0 und Alt1 bis Alt0 für 11 bis 20 gedrückt werden. ⇧ Shift zu drücken um mehrere Ebenen auszuwählen funktioniert hier ebenfalls.

Mehrere Ebenen

Ein Objekt kann auf mehreren Ebenen existieren, zum Beispiel kann eine Lampe auf den Ebenen 1,2 und 3 würde ein Objekt auf den Ebenen 3 und 4 aber nicht eins auf den Ebenen 4 und 5 beleuchten. Es gibt viele nützliche Anwendungen hierfür, vor allem bei Beleuchtung und Partikelsystemen.

Ein Objekt zwischen Ebenen bewegen



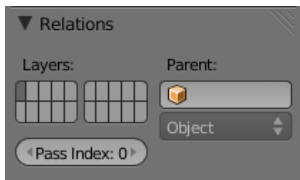
Ebenen Auswahl

Um ausgewählte Objekte auf eine andere Ebene zu verschieben drücke M und wähle die gewünschte Ebene aus dem pop-up Dialog.



Objekt kontext auswahl

Ein anderer Weg um die Ebenen eines Objektes anzusehen oder zu bearbeiten ist über das Beziehungen Panel im Objekt kontext.



Ebenen im Objekt kontext,
Beziehungen Panel.

Dort gibt es wieder die verschiedenen Ebene aus denen man die auswählen kann auf denen das Objekt existieren soll.

Animations Ebenen

Die zugehörigkeit eines Objektes zu einer Ebene kann animiert werden um das Auftauchen oder Verschwinden von Objekten zu ermöglichen.

Beispiel Ebenen Organisation

Die Oberen Ebenen könne für die wichtigen Teile der Szene genutzt werden und die unteren für die die nicht so häufig geändert werden. In einer Szene mit Zwei Charakteren könnten die Objekte auf den Ebenen folgendermaßen organisiert werden:

1. Hauptdarsteller
2. Nebenschauspieler
3. Hintergrundschauspieler
4. Partikel und Effekte
5. Hauptbühne
6. Haupthintergrund
7. Hauptrequisiten
8. kleinere Requisiten und Dekorationen
9. Kamera und Beleuchtung
10. Die Armaturen der Hauptdarsteller
11. Die Armaturen der Nebenschauspieler
12. Die Armaturen der Hintergrundschauspieler
13. alternative Kleidung
14. Meshes die noch in Arbeit sind
15. Andere Bühnenaufteilung
16. alternative Hintergründe
17. andere Dekorationen
18. Dekorationen die noch in Arbeit sind
19. Zusätzliche Beleuchtung

lokale oder globale Ansicht

Mode: All modes

Hotkey: / NumPad

Menu: View » Local View or View » Global View

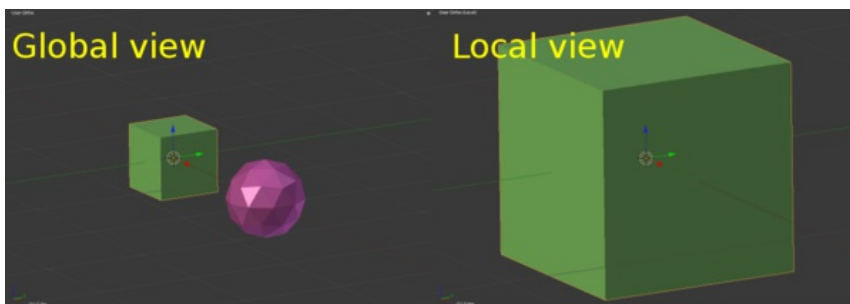
Beschreibung

In der lokalen Ansicht werden nur ausgewählte Objekte dargestellt, was Szenenbearbeitung einfacher machen kann. Um die lokale Ansicht zu benutzen sind zuerst die gewünschten Objekte auszuwählen und dann View » Local View zu benutzen. Benutze View » Global View um zur globalen Ansicht zurückzukehren. Alternative kann zum hin- und herschalten / NumPad benutzt werden.

Es ist zu beachten das die Ebenen Buttons in der lokalen Ansicht verschwinden.

Beispiele

In der globalen Ansicht sind alle Objekte sichtbar. Wird der grüne Würfel ausgewählt und in die lokale Ansicht gewechselt, wird der Würfel in der 3D Ansicht zentriert. Falls eine Szene viele Objekte enthält kann es den Arbeitsprozess enorm beschleunigen das nur ausgewählte Objekte sichtbar sind.



globale und lokale Ansicht

Transformationen

Es gibt viele Dinge die sie mit den ausgewählten Objekten in Ihrer virtuellen Welt anstellen können. Sie können sie drehen, größer oder kleiner machen, dessen Form ändern, und so weiter. Diese Abteilung erklärt Ihnen wie Sie diese Dinge auf Ihre Objekte anwenden können.

Hier ist eine Liste an verfügbaren Transformationen::

- - [Mirror](#)
 - [Push Pull](#)
 - [Shear](#)
 - [To Sphere](#)
 - [Warp](#)
- [Basics](#)
 - [Grab](#)
 - [Rotate](#)
 - [Scale](#)

Mittels dieser Tastenkombinationen können Sie Transformationen anwenden:

- **shortcuts**

Tippen Sie auf eine Taste um eine Aktion auszuführen

- **context-sensitive menus**

Mit einem ausgewählten Objekt in der 3D Ansicht, im With an object selected in a 3D view, in Objekt Modus, zeigt die Menüleiste die Ansicht, Auswählen und Mesh Optionen. Klicken Sie auf die Mesh Optionen um das Objekt zu manipulieren one to manipulate the object. Das Menü das erscheint hat die Optionen Transformieren. Über der Option Transformieren erscheint ein Unter-Menü, dass Ihnen Optionen anzeigt (und die Tasten zum Aufrufen dieser auf der rechten Seite)) zum Manipulieren des Objekts.

Grundlegende Manipulationen

Dieser Abschnitt fasst alle grundlegenden Transformationen und Manipulationen in den 3D-Ansichten zusammen:

- [Greifen \(Bewegen\) von Elementen.](#)
- [Rotieren von Elementen.](#)
- [Skalieren von Elementen.](#)
- [Elemente bei Transformationen an etwas anheften \(Geometrie, Gitter, usw.\).](#)
- [Präzise Kontrolle von Elementen.](#)
- [Durchführen von Transformationen durch das Eintippen von Zahlen anstelle von Maus-Aktionen.](#)
- [Zurücksetzen von Objekt-Transformationen.](#)

Greifen (Bewegen) von Elementen

Mode: Object und Edit Mode

Hotkey: G

Menu: Object/Mesh/etc. → Transform → Grab/Move

Beschreibung

Einer der schnellsten Wege, Gegenstände im 3D-Raum zu bewegen, steht mit G zur Verfügung. Durch Drücken dieser Hotkey-Taste wird der Bearbeitungsmodus "Grab/Move" ("Fassen/Bewegen") eingeschaltet, in dem ausgewählte Objekte oder Daten entsprechend der Mauszeigerposition bewegt werden können. Dabei spielt die Entfernung zwischen Mauszeiger und dem zu bearbeitenden Objekt keine Rolle.

Optionen

LMB 

Bestätigt die Bewegung und setzt das Objekt oder die Daten an ihrer momentanen Position auf dem Bildschirm ab.

MMB 

Begrenzt die Bewegung auf die X-, Y- oder Z-Achse.

RMB  oder Esc

Abbrechen der Bewegung, Rückkehr des Objekts oder der Daten an ihre Ausgangsposition.

Rotieren von Elementen

Mode: Object und Edit Mode

Hotkey: R



Menu: Object/Mesh/etc. → Transform → Rotate

Einführung

Rotationen sind auch bekannt unter [den im Englischen synonymen Begriffen] spin, twist, orbit, pivot, revolve oder roll. In der Blender-Dokumentation wird aus Gründen der Eindeutigkeit nur von "Rotation" oder "Rot" die Rede sein. Die vier Wege, ein Objekt zu rotieren, sind:

1. Das Tastaturkürzel (**die schnellste, von den meisten Benutzern gewählte Methode**)
2. (TODO) Der Mausmanipulationen-Widget (einfach)
3. (TODO) Das Properties-Menü (einfach)
4. (TODO) Ein Python-Script (flexibel und interessant)

Rotation mittels Tastaturkürzel

1. RMB  zur Auswahl des Objekts
2. Einmal R drücken, um in den Rotationsmodus zu gelangen.
3. Das Objekt mittels Mausbewegungen rotieren. Je näher sich die Maus am Objektmittelpunkt befindet, desto größer ist der Einfluss auf die Rotation.
4. LMB  drücken, um die Änderungen zu bestätigen.

Sperren von Rotationsachsen

Ein Objekt lässt sich leicht um die **X-Achse** drehen, wenn man nacheinander R und X drückt.

Mehrere verschiedene Achsen können auch gesperrt werden:

R, X : Rotation entlang der **X-Achse**

R, Y : Rotation entlang der **Y-Achse**

R, Z : Rotation entlang der **Z-Achse**

Um entlang von zwei Achsen unter Auslassung einer dritten zu rotieren, benutzt man die Hochstelltaste (Shift):

R, ⇧ Shift+X: Rotation entlang der **YZ-Achsen**

R ⇧ Shift+Y: Rotation entlang der **ZX-Achsen**

R ⇧ Shift+Z: Rotation entlang der **XY-Achsen**

Feinabstimmung einer Rotation

Snapping (Einrasten)

Taste Ctrl während einer Rotation gedrückt halten, um die Rotation an der nächsten Gradzahl einrasten zu lassen.

Tweaking (Kleinstschrittiges Arbeiten)

Taste ⇧ Shift gedrückt halten, um langsam in Schritten von hundertstel Grad zu rotieren.

Trackball Rotation (Trackball-Rotation)

Drücken von R,R, um den Rotationsmodus zu ändern und die Rotation aus einer lokalen Perspektive aus vorzunehmen.

Skalieren von Elementen

Mode: Object und Edit Mode

Hotkey: S

Menu: Object/Mesh/etc. → Transform → Scale

Beschreibung

Durch Drücken von S wird der Bearbeitungsmodus "Scale" ("Skalieren") aufgerufen, in dem ausgewählte Objekte oder Daten in bezug auf die Position des Mauszeigers verkleinert oder vergrößert werden können. Die Objekt- bzw. Datengröße wird vergrößert, wenn der Mauszeiger vom Drehpunkt (Pivot Point) wegbewegt wird, und wird verkleinert, wenn der Mauszeiger auf diesen zubewegt wird. Wechselt der Mauszeiger auf die andere Seite des Drehpunkts, wird die Skalierung in negativer Richtung fortgesetzt, wodurch das Objekt bzw. die Daten umgedreht erscheint. Die Genauigkeit der Skalierung wird von der Entfernung des Mauszeigers vom Objekt bzw. den Daten bei Beginn der Skalierung bestimmt.

Optionen

LMB 

Bestätigt die Skalierung und belässt das Objekt oder die Daten in ihrer augenblicklichen Größe auf dem Bildschirm.

MMB 

Beschränkt die Skalierung auf die X-, Y- oder Z-Achse.

RMB  or Esc

Abbrechen der Skalierung und Zurücksetzen des Objekt oder der Daten auf die Ausgangsgröße.

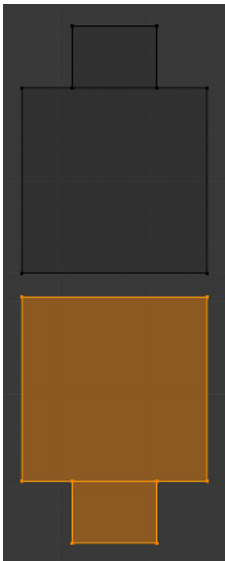
Spiegeln

Mode: Object und Edit Modi

Hotkey: CtrlM

Menu: Object/Mesh » Mirror

Beschreibung



eine Auswahl spiegeln

Eine Objekt oder Mesh Auswahl zu Spiegeln erzeugt eine umgekehrte Version der Auswahl. Die Position der gespiegelten Version der Auswahl hängt vom [Pivot Punkt](#) ab. Eine gewöhnlich Anwendung des Spiegeln wäre eine Hälfte eines Objekts zu formen, sie zu duplizieren und dann zu spiegeln um eine umgekehrte Version zum vervollständigen des Modells zu erhalten. Spiegelungen können auch mit dem [Spiegel Modifier](#) erzeugt werden.

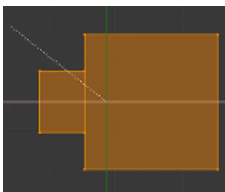
Benutzung

Um eine Auswahl an einer globalen Achse zu spiegeln drücke:

StrgM, gefolgt von X, Y oder Z.

Das Bild eine Auswahl spiegeln zeigt das Ergebnis einer solchen Operation nach dem ein Meshteil dupliziert wurde.

[Lerne mehr über Transformationen](#)



Interactives spiegeln

Alternativ kann die **MMB**  gedrückt werden um das Objekt interaktiv mit bewegungen zur Spiegelachse zu spiegeln.

Kugelförmig

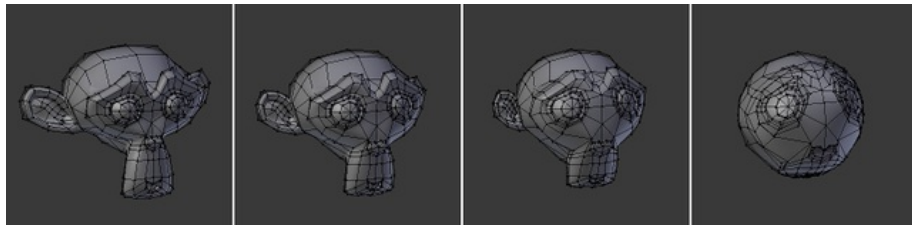
Mode: Bearbeitungs Modus

Hotkey: ⇧ ShiftAltS

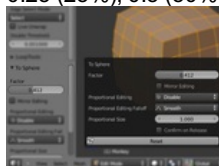
Menu: Mesh » Transformieren » Kugelförmig

Beschreibung

Die kugelförmige Transformation nähert die Auswahl an die Form einer Kugel an. Das *Suzanne mit sich verstärkender Kugelform* Bild unten zeigt das Ergebnis einer solche Transformation.



Suzanne mit sich verstärkender Kugelform. Diese Sequenz zeigt ein Suzanne Mesh mit 0, 0.25 (25%), 0.5 (50%) und 1 (100%) einer kugelförmigen Transformation.

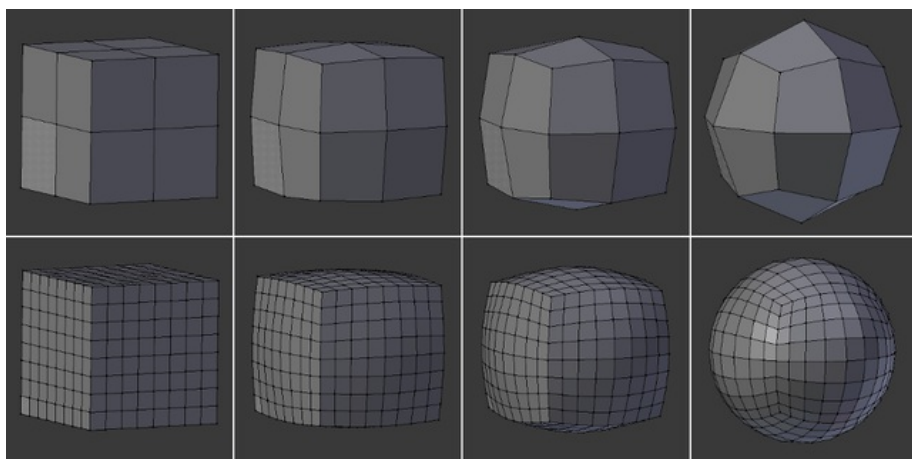


Kugelförmigkeits
Faktor

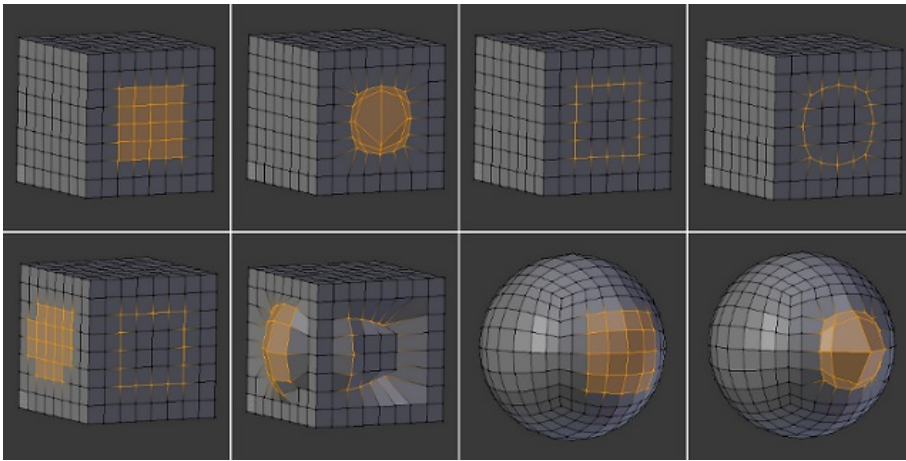
Benutzung

Die zu verändernden Elemente sind auszuwählen und kugelförmige Transformation ist anzuwenden. Dieser Befehl kann vom Mesh » Transformieren » Kugelförmig Menü oder durch das drücken von ⇧ ShiftAltS aufgerufen werden. Das Ausmaß an Kugelförmigkeit kann durch bewegen der Maus oder Zahleneingabe eingestellt werden. ↵ Enter zu drücken schließt die Operation ab. Die Operation kann danach allerdings mit F6 oder durch das Bedienen der Funktionen im Toolshelf geändert werden, wenn noch keine andere Operation ausgeführt wurde.

Es ist zu beachten das das Ergebnis der kugelförmigen Transformation von der Anzahl der ausgewählten Mesh-elemente abhängt. Wie im Bild unten zu sehen ist, wird das Ergebnis runder und Kugelförmiger je mehr Teile des Meshes zum Transformieren zur Verfügung stehen.



kugelförmige Transformation angewendet auf Würfel mit unterschiedlichen Unterteilungstufen. In dieser Sequenz wurde die kugelförmige Transformation auf den ganzen Würfel angewendet mit den Faktoren 0, 0.25 (25%), 0.5 (50%) und 1 (100%) .



kugelförmige Transformation angewendet auf unterschiedliche Auswahlen

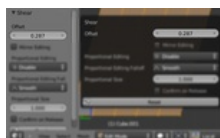
Verschieben

Mode: Object and Edit modes

Hotkey: ⇧ ShiftCtrlAltS

Menu: Object/Mesh/Curve/Surface » Transform » Shear

Beschreibung



Verschieben Faktor

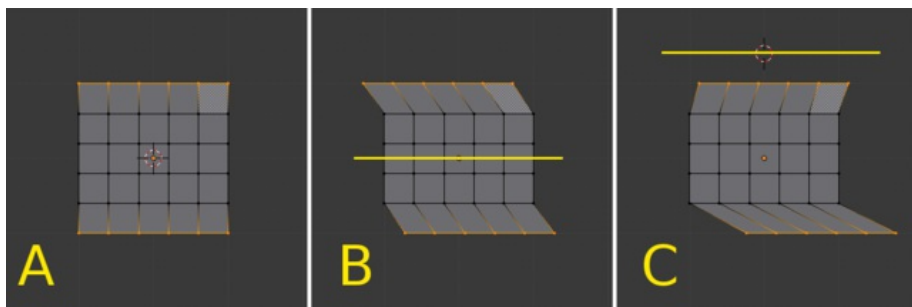
Verschieben ist eine Form der Bewegung bei der zwei parallele Ebenen sich parallel aneinander vorbei bewegen. Bei dieser Transformation wird die Verschiebung parallel zur Horizontalen der aktuellen Sicht durchgeführt. Die Achsen Position wird durch den Pivot Punkt festgelegt. Alles über der Achse wird in die gleiche Richtung wie der Mauszeiger bewegt werden, alles darunter in die entgegengesetzte Richtung.

[Lerne mehr über Pivot Punkte »](#)

Benutzung

Wähle die gewünschten Elemente aus und aktiviere die Verschieben Transformation. Die Verschieben Funktion kann auch mit dem MenüObject/Mesh/Curve/Surface » Transform » Shear oder durch drücken von ⇧ ShiftCtrlAltS aufgerufen werden. Das Ausmaß an Bewegung kann durch Mausbewegungen oder Zahleneingabe bestimmt werden. ↵ Enter zu drücken schließt die Operation ab. Die Operation kann danach allerdings mit F6 oder durch das Bedienen der Funktionen im Toolshelf geändert werden, wenn noch keine andere Operation ausgeführt wurde.

Es ist zu beachten das das Ergebnis der Verschieben Transformation von der Anzahl der ausgewählten Mesh-elemente abhängt. Unten ist zu sehen wie sich Verschieben auf eine Anzahl von Elementen auswirkt.



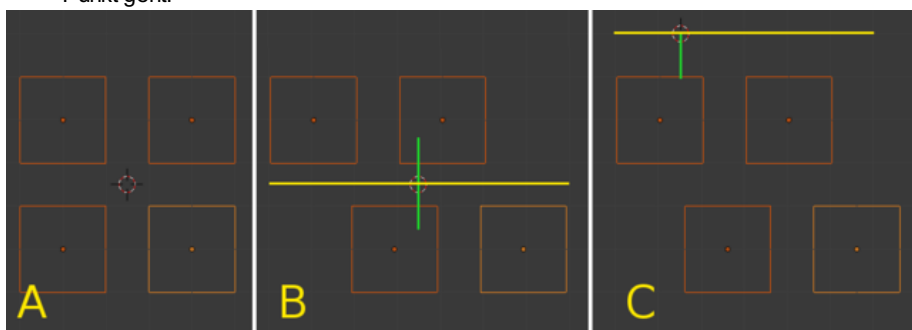
Die Effekte der Verschieben Transformation mit unterschiedlichen Pivot Punkten. Siehe Text für mehr Information

Die drei Bilder oben zeigen die Auswirkungen des Verschiebens ausgewählter Vertices wenn der Pivot Punkt verändert wird. In Bild B ist der Pivot Punkt auf Median Point eingestellt und die Maus wurde während der Transformation nach links bewegt. In Bild C ist der Pivot Punkt auf den 3D Cursor eingestellt der über dem Mesh positioniert ist. Wenn die Maus nach links bewegt wird werden die ausgewählten Vertices nach rechts bewegt da sie unter dem Pivot Punkt liegen.



Ausmaß des Verschiebens

Das Ausmaß des Verschiebens ist linear also direkt proportional zur Entfernung des Punkts von der Achse die durch den Pivot Punkt geht.



Die Effekte einer Verschiebung auf Objekte mit unterschiedlichen Pivot Punkten. Siehe den Text unter für mehr Information.

Die drei Bilder oben zeigen die Auswirkungen des Verschiebens ausgewählter Objekte wenn der Pivot Punkt verändert wird. In Bild B ist der Pivot Punkt auf Median Point eingestellt und die Maus wurde während der Transformation nach links bewegt. In Bild C ist der Pivot Punkt auf den 3D Cursor eingestellt der über den Objekten positioniert ist. Wenn die Maus nach links bewegt wird werden die ausgewählten Objekte nach rechts bewegt da sie unter dem Pivot Punkt liegen.

Wölben

Mode: Object and Edit modes

Hotkey: ⇧ ShiftW

Menu: Object/Mesh/Curve/Surface » Transform » Warp

Beschreibung



Wölbungs Winkel

Im Edit Modus, transformiert die Wölbung die ausgewählten Meshelemente und wölbt sie um den 3D Cursor in einem bestimmten Winkel. Diese Transformations ist allerdings immer abhängig von der Position des 3D Cursors und vom Sichtpunkt ab.

Im Objekt Modus, transformiert die Wölbung die ausgewählten Objekte und wölbt sie um den 3D Cursor in einem bestimmten Winkel. Diese Transformations ist allerdings immer abhängig von der Position des 3D Cursors und vom Sichtpunkt ab.

Benutzung

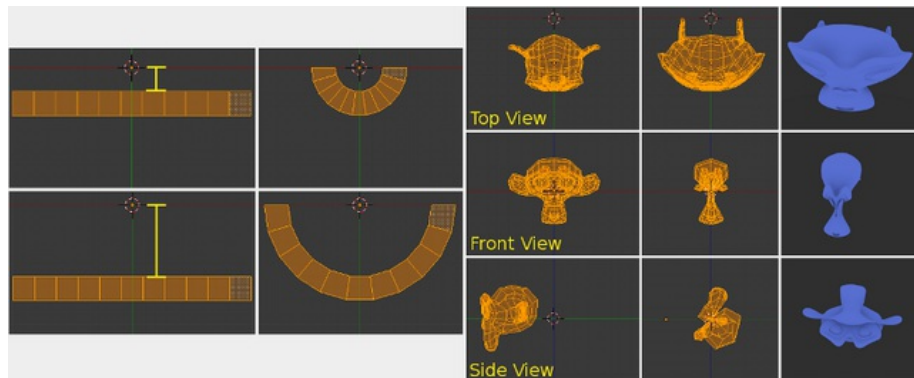
Wähle die gewünschten Elemente aus und aktiviere die Wölben Transformation. Die Verschieben Funktion kann auch mit dem Menü Object/Mesh/Curve/Surface » Transform » Warp} oder durch drücken von ⇧ ShiftW aufgerufen werden. Das Ausmaß an Bewegung kann durch Mausbewegn oder Zahleneigabe bestimmt werden. ↵ Enter zu drücken schließt die Operation ab. Die Operation kann danach allerdings mit F6 oder durch das Bedienen der Funktionen im Toolshelf geändert werden, wenn noch keine andere Operation ausgeführt wurde.

[[File:3D-interaction_Transformation_Advanced_Warp_warp-mesh.png|center|frame|In diesem Beispiel wird eine Ebene um den 3D Cursor mit den agegebenen Wineklgrößen gewölbt.

Cursor- und Sichtabhängigkeit

Die Position des 3D Cursors kann das Ergebnis der Wölbung beeinflussen, wie im Beispiel zu sehen ist. Der Wöblungsradius ist abhängig von der Entfernung des Cursors zu den ausgewählten Elementen.

Das Ergebnis der Wölbung hängt auch vom Sichtpunkt ab, in den Beispielen wurde immer eine 180° Wölbung auf das Suzanne Mesh angewendet, aber immer mit anderen Ansichten.



Das linke Bild zeigt eine Wölbung die vom Cursor beeinflusst wurde, das rechte zeigt den Einfluss des Sichtpunkts.

Logo Beispiel

Eine Verwendung für diese Werkzeug ist das Wölben eines Texts um ein Logo. Das Bild unten wurde erzeugt aus dem Blender Logo und dem Text als separate Objekte. Der Text wurde in ein Mesh umgewandelt und um das Blender Logo gewölbt.



Text Wölben

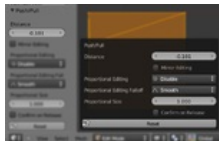
Zum transformieren von Text muss dieser erst mit AltC oder durch auswählen von Mesh from Curve/Meta/Surf/Text in ein Mesh umgewandelt werden.

Push/Pull

Mode: Objekt und Edit Modi

Menu: Object/Mesh » Transform » Push Pull

Beschreibung



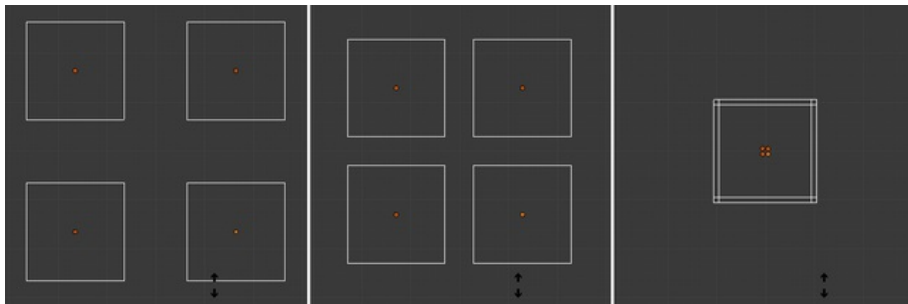
Push/Pull Entfernung.

Push/Pull verschiebt die ausgewählten Elemente (Objekte, Vertices, Kanten oder Flächen) um die gleiche Entfernung näher zusammen oder auseinander.

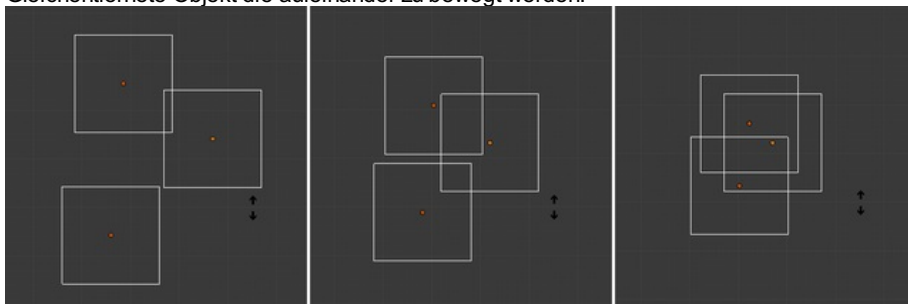
Benutzung

Wähle die gewünschten Elemente aus und aktiviere die Push/Pull Transformation. Die Push/Pull Funktion kann auch mit dem Menü Object/Mesh/Curve/Surface » Transform » Push/Pull oder durch drücken von $\text{⇧ Shift} + \text{A}$ aufgerufen werden. Das Ausmaß an Bewegung kann durch Mausbewegung oder Zahleneingabe bestimmt werden. ↵ Enter zu drücken schließt die Operation ab. Die Operation kann danach allerdings mit F6 oder durch das Bedienen der Funktionen im Toolshelf geändert werden, wenn noch keine andere Operation ausgeführt wurde.

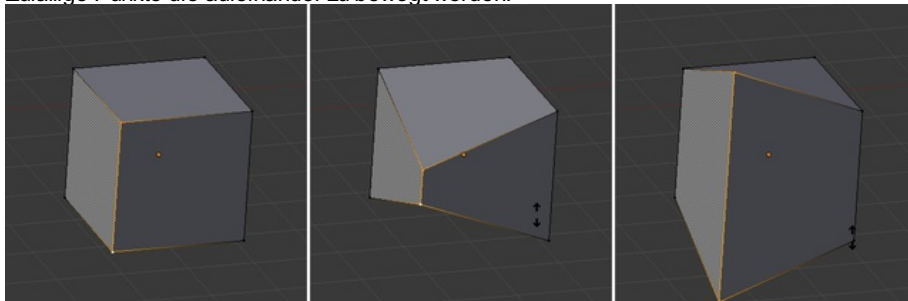
Dabei ist zu beachten das die Push/Pull Transformation von der Anzahl und dem Typ der ausgewählten Objekte abhängt. Siehe unten für die Ergebnisse von Push/Pull auf verschiedene Elemente.



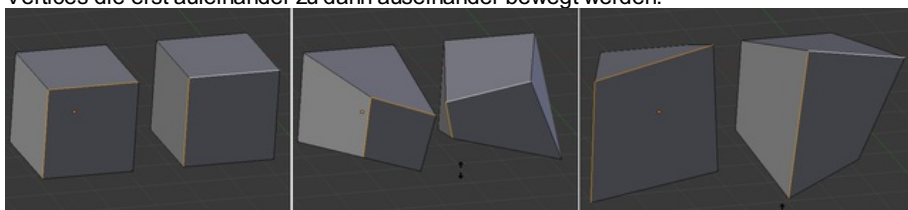
Gleichferne Objekte die aufeinander zu bewegt werden.



Zufällige Punkte die aufeinander zu bewegt werden.



Vertices die erst aufeinander zu dann auseinander bewegt werden.



Kanten verschiedener Meshes die erst aufeinander zu dann auseinander bewegt werden.

Transformationskontrollen

Transformationskontrollen werden benutzt um die verfügbaren [Transformationen](#) besser zu kontrollieren.

Die folgenden Seiten detaillieren die verfügbaren Kontrolloptionen:

Transformations Ausmaß

- [Präzision der Transformation](#)
- [Zahleneingabe](#)
- [Transformations Eigenschaften](#)
- [Objekttransformationen zurücksetzen](#)
- [Proportionales Bearbeiten](#)

Transformations Orientierung

- [Manipulators](#)
- [Transformations Orientierungen](#)
- [Achsenfestlegung](#)

Transformationszentrum

- [Pivot Punkt](#)
- [aktive Objekte](#)
- [individuelle Zentren](#)
- [3D Cursor](#)
- [Median Punkt](#)
- [Bounding Box Zentrum](#)

Transformation "Springen"

- ["Springen" Transformationen](#)
- [An Mesh springen](#)

Präzision

Mode: Object and Edit modes

Hotkey: Ctrl and/or ⇧ Shift

Beschreibung

Das halten von Strg oder ⇧ Shift während einer Transformation erlaubt eine Präzisierung der Aktion auf feste und oder sehr kleine Werte, die im 3D Ansicht Balken links zu sehen sind.

[Lerne mehr über die Fensterbalken. »](#)

mit Transformationsmanipulatoren

Das funktioniert auch mit den [Transformations Manipulatoren](#) im 3D Fenster.

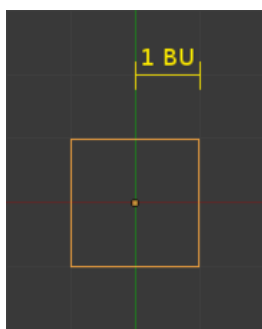


Kombinieren mit anderen Kontrollen

Alle hier beschriebenen Kontrollen können mit Kontrollen über [Achsenfestlegung](#) und [Pivot Punkte](#) kombiniert werden.

Strg halten

Bewegungstransformationen



Eine Blender Einheit wie man sie unter Standardvergrößerung sehen kann.

Bei Bewegungstransformationen bei Standardvergrößerung verursacht das Halten von Strg eine Verschiebung um eine Blender Einheit. Wenn man in orthografischer sicht weit genug vergrößert das man die nächst kleinere oder Unterteilung sieht, ist die Schrittgröße verzehnfacht oder gezehntelt.

[Lerne mehr über Vergrößerung. »](#)

Rotations Transformationen

Bei Rotationen bewirkt das Halten von Strg rotation in 5° Schritten.

Skalierungs Transformationen

Beim Skalieren bewirkt das halten von Strg skalieren in Faktorstufen von 0,1.

Spring Modi

Beachte das, wenn [Springe zu Element](#) eingeschaltet ist, das Halten von Strg das Springen zum nächsten Element verursacht. [Lerne mehr über das Springen von Elementen »](#)

⇧ Shift halten

Das Halten von ⇧ Shift während Transformationen erlaubt sehr feine Kontrolle über die Bewegungsausmaße.

STRG und SHIFT halten

Bewegungstransformationen

Bei Bewegungstransformationen bei Standardvergrößerung verursacht das Halten von Ctrl⇧ Shift eine Verschiebung um 1/10 einer Blender Einheit. Vergrößerungen spielen keine Rolle.

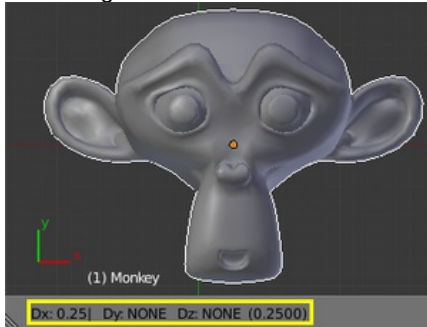
Rotations Transformationen

Das Halten von Strg⇧ Shift beim rotieren bewirkt Rotationen um 1°.

Skalierungstransformationen

Das Halten von Strg⇧ Shift beim skalieren bewirkt veränderungen in Schritten der Größe 0,01 Blender Einheiten.

Zahleneingabe



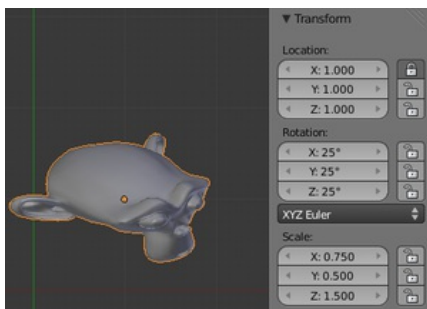
Zahleneingabe im 3D Ansichtsbalken

Die Maus für Transformationen zu benutzen ist zwar intuitiv aber ungenau. Es können Zahlen für alle Transformationen eingegeben werden um das Ausmaß exakt einzustellen. Dazu muss erst die gewünschte Funktion aufgerufen werden, dann können Zahlen eingegeben werden.

[Lerne mehr über Transformationsausrichtung »](#)

[Lerne mehr über Achsenfeslegung »](#)

Eingabe über die Werkzeugbalken



Transformationsausmaße können auch über das Toolshelf eingegeben werden.

Transformations Ausmaße können auch im Toolshelf eingestellt werden.

transformatorische Eigenschaften

Jedes Objekt hat Positions, Ausrichtungen und Skalierungs Werte, die einfach eingestellt oder verändert werden können.

Panel für transformatorische Eigenschaften

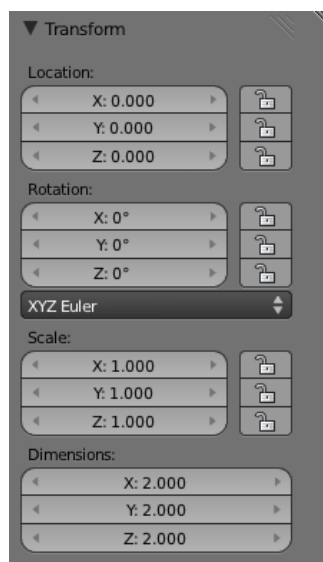
Mode: Edit and Object modes

Hotkey: N

Menu: Object » Transform Properties

Der Teil für transformatorische Eigenschaften des Eigenschaften Panels erlaubt das einfache ansehen und verändern von Position, Rotation und Skalierung von Objekten im Objekt Modus und die genaue Einstellung von Positionen, lokal oder global, für Vertices oder die Zentren von Verticegruppen im Edit Mode. [Lerne mehr über Modellierung.](#)

Optionen im Objekt Modus



transformatorisches
Eigenschaften Panel im Objekt
Modus.

Location X, Location Y, Location Z
Die Position des Objekts in 3D Koordinaten.

Rotation X, Rotation Y, Rotation Z
Die Ausrichtung des Objekts zu den globalen Achsen.

Scale X, Scale Y, Scale Z
Die Skalierung des Objekts entlang der lokalen Koordinaten.

Dimensions X, Dimensions Y, Dimensions Z
Die Ausmaßen des Objekts in Blender Einheiten entlang der Achsen.

Ipo Note

Die Werte für Ort, Rotation und Skalierung können mit IPOunkten animiert werden. Falls ein IPOpunkt mit dem Objekt verbunden ist, muss die neue Einstellung als neuer IPOpunkt gespeichert werden um sie zu erhalten.

Für mehr Informationen wie man Werte verändert, siehe das [Interface](#).

transformatorische Eigenschaften sperren

Das Sperren von transformatorischen Eigenschaften legt fest das diese Eigenschaften nur noch im entsprechenden Panel eingestellt werden können. Das bedeutet das Transformationen die entsprechende Eigenschaft nicht mehr verändern können.

Um ein Feld zu sperren klicke auf das offene , um es zu öffnen auf das geschlossene  Schloßsymbol neben dem Feld.

eingestellte Transformationen anwenden

Die eingestellten Transformationen anzuwenden setzt die Werte des Objekts auf die Standardwerte (0,0,0) für die Position und Rotation und (1,1,1) für die Skalierung zurück und wendet alle vorher eingestellten Werte auf enthaltene Meshes und Objekte an. Ein

vorher auf Faktor 4 hochskaliertes Objekt ist danach viermal so groß wie das Original, hat aber wieder eine Skalierung von 1.

Um eine Transformation anzuwenden wähle Objekt>>anwenden oder StrgA und wähle die entsprechende Transformationseigenschaft aus.

Make Duplicates Real löst Verknüpfungen von verlinkten Duplikaten sodass diese nun unabhängige Datenblöcke haben und Transformationen am Ursprungsobjekt keine Auswirkungen mehr haben.

Transformationen zurücksetzen

Transformationen zurücksetzen setzt alle Werte auf die Standardwerte zurück. Dieser Befehl ist für die verschiedenen Eigenschaften separat verfügbar mit:

- setze Position zurück AltG
- setze Rotation zurück AltR
- setze Skalierung zurück AltS
- setze Ursprung zurück AltO

Objekt Transformationen entfernen

Modus: Objektmodus

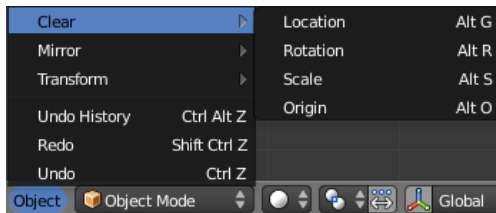
Tastenkombination: AltG, AltS, AltR, AltO

Menü: Objekt » Entfernen » Position / Rotation / Skalieren / Ursprung

Beschreibung

Clearing transforms simply resets the transform values. The objects location and rotation values return to 0, and the scale returns to 1.

Optionen zum Entfernen



Clear Transformation menu

Position entfernen AltG

Entfernen (zurücksetzen) der Position der aktuellen Auswahl. Das wird die Auswahl zurück auf die Koordinaten 0, 0, 0 verschieben.

Skalierung entfernen AltS

Entfernen (zurücksetzen) der Skalierung der aktuellen Auswahl. This will resize the selection back to the size it was when created.

Rotation entfernen AltR

Entfernen (zurücksetzen) der Rotation der aktuellen Auswahl. This will set the rotation of the selection to 0 degrees in each plane.

Ursprung entfernen AltO

Entfernen (zurücksetzen) des Ursprungs der Kinderobjekte. This will cause Child objects to move to the coordinates of the parent.

Anwenden von Objekt Transformationen

Modus: Objektmodus

Tastenkombination: StrgA

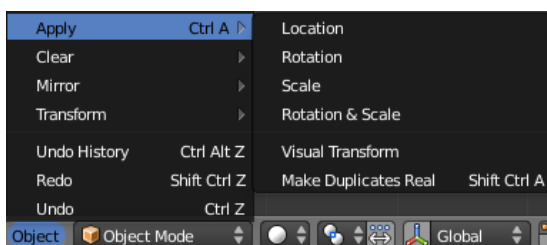
Menü: Objekt » Anwenden

Applying transform values essentially resets the values of object's position, rotation, or scale, but does not actually do anything to the object. The center point is moved to the origin and the transform values are set to zero. In terms of scale, the scale values return to 1.

To apply a transform select the Apply sub-menu from the Object menu or use the shortcut CtrlA and select the appropriate transform to apply

Duplikate real machen unlinks linked duplicates so each duplicate now has its own datablock.

Optionen zum Anwenden



Apply Transformation menu

Apply Location CtrlA

Apply (set) the location of the selection. This will make Blender consider the current location to be equivalent to 0 in each plane i.e. the selection will not move, the current location will be considered to be the "default location". The Object Center will be set to actual 0,0,0 (where the coloured axis lines intersect in each view).

Apply Rotation CtrlA

Apply (set) the rotation of the selection. This will make Blender consider the current rotation to be equivalent to 0 degrees in each plane i.e. the selection will not be rotated, the current rotation will be considered to be the "default rotation".

Apply Scale CtrlA

Apply (set) the scale of the selection. This will make Blender consider the current scale to be equivalent to 0 in each plane i.e. the selection will not be scaled, the current scale will be considered to be the "default scale".

Apply Rotation and Scale CtrlA

Apply (set) the rotation and scale of the selection. Do the above two applications simultaneously.

Apply Visual Transform CtrlA

Apply (set) the result of a constraint and apply this back to the Object's location, rotation and scale. See the following post for more detailed discussion: [Apply visual transform](#).

Make Duplicate Real ⇧ ShiftCtrlA

Make any duplicates attached to this Object real so that they can be edited.

Manipulatoren

Mode: Object and Edit modes


Hotkey: CtrlSpace

In Verbindung mit [Achsenfestlegung](#) können die [grundlegenden Transformationsbefehle](#) benutzt werden um Objekte an alle Achsen zu verschieben. Es gibt aber Situationen wo das nicht sinnvoll ist. Zum Beispiel wenn eine Fläche eines zufällig rotierten Objekts in eine Richtung senkrecht zu seiner Normalen verschoben werden soll. Dafür gibt es die Transformations Manipulatoren.



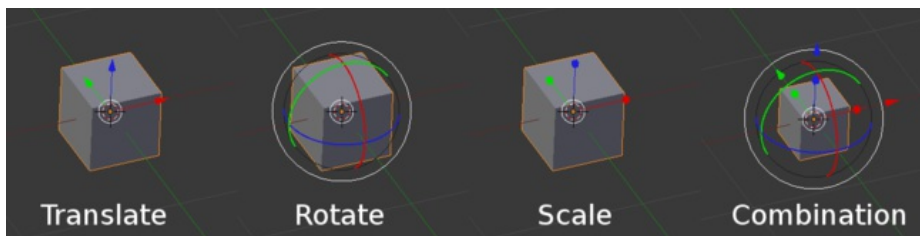
Der Abschnitt der Transformationsmanipulatoren im 3D Ansichtsbalken.

Transformationsmanipulatoren stellen eine visuelle Repräsentation der Transformationsoperationen zu Verfügung. Sie können ein- und ausgeschaltet werden durch klicken auf das entsprechende Bild oder über den Schnellzugriff CtrlSpace.

Es gibt einen Manipulator für jeden Transformations Befehl. Sie können allein oder in Kombination (⇧ Shift LMB  auf die verschiedenen Pfeile und Bögen) benutzt werden.

Manipulators can be accessed in the header of the 3D View window:

- Achse: Ein- und Ausschalten der Manipulatoren.
- Pfeil: Translation.
- Bogen: Rotation.
- Box: Skalierung.
- Transformausrichtungsmenü: Wahl der Transformationsausrichtungsart.



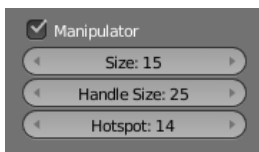
Manipulatoroptionen

Manipulatorkontrollen

Im Rotationsmodus steht der weiße Kreis für Rotation um die Sichtrichtung und steht für die gleiche Funktion wie R. Wiederholtes drücken von R und der graue Kreis stehen für den "Trackball" Rotationsmodus.

Lerne mehr über [Präzision](#), [Achsenfestlegung](#) bei Transformationen und die ["Trackball" Rotation](#).

Manipulator Benutzereinstellungen



Manipulator Benutzereinstellungen

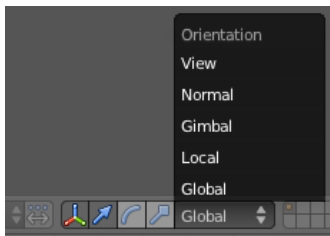
Die Einstellungen für die Manipulatoren kann man im Interface Bereich der Benutzereinstellungen festlegen.

- Größe: Durchmesser der Manipulatoren in 10 Pixel Einheiten.
- Handlesize: Größe der Manipulatorgriffe als Prozent des Durchmessers (Size/2).
- Hotspot: Größe der klickbaren Fläche um die Manipulatoren anzuwenden.

Auswahl des Vergleichskoordinatensystems

Mode: Object and Edit modes

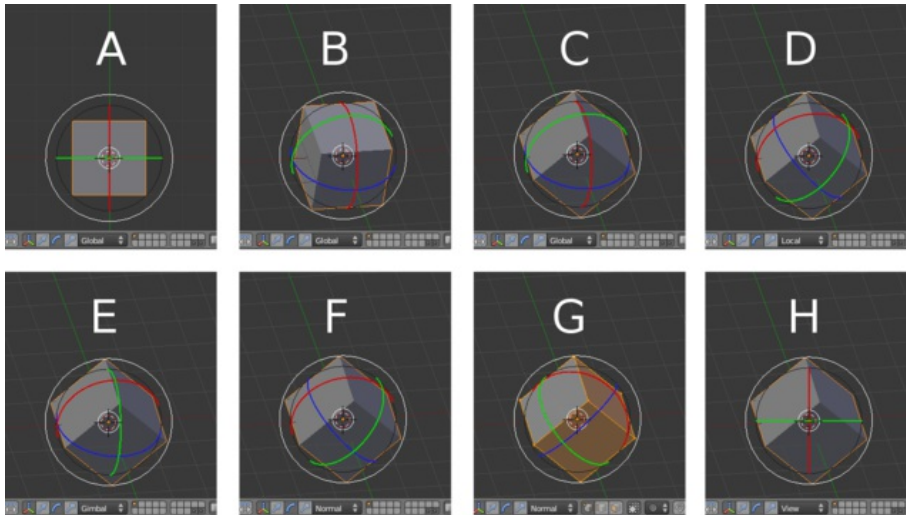
Hotkey: AltSpace



Auswahl des
Vergleichskoordinatensystems

Das [Vergleichskoordinatensystem](#) kann auf global, lokal, gimbal, normal oder von der Sichtachse aus eingestellt werden. Das Bild unten zeigt die verschiedenen Einstellungen von A bis F.

Die normale Einstellung ändert ihre Funktion Kontextbezogen. Sie steht für die zusammengerechnete Normalen aller ausgewählten Objekte.



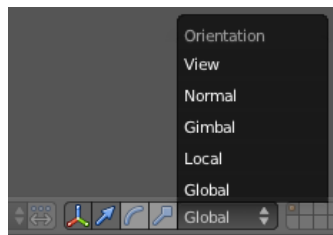
Vergleichskoordinatensysteme für die Transformationen.

- A: Standard Würfel mit *globalem* Vergleichskoordinatensystem ausgewählt.
- B: Gedrehter standard Würfel mit *globalem* Vergleichskoordinatensystem ausgewählt.
- C: Gedrehter standard Würfel mit *globalem* Vergleichskoordinatensystem ausgewählt.
- D: Gedrehter standard Würfel mit *lokalem* Vergleichskoordinatensystem ausgewählt.
- E: Gedrehter standard Würfel mit *gimbalem* Vergleichskoordinatensystem ausgewählt.
- F: Gedrehter standard Würfel mit *normalem* Vergleichskoordinatensystem ausgewählt.
- G: Gedrehter standard Würfel mit *normal* Vergleichskoordinatensystem mit einigen Vertices ausgewählt.
- H: Gedrehter standard Würfel mit *view* Vergleichskoordinatensystem ausgewählt.

Ausrichtungen zur Transformation

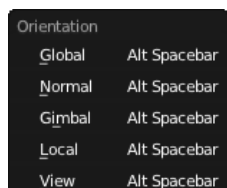
Modus: Objektmodus und Editiermodus

Tastenkombination: AltLeertaste



Transformationsausrichtung's Auswahl menu.

Ausrichtungen beeinflussen das Verhalten von Transformationen: Position, Rotation und Rotation. Ein Effekt dessen ist auf dem 3D Manipulator zu finden (das Widget im Zentrum Ihrer Auswahl), und auf Transformations Beschränkungen (siehe [Achsen-Beschränkung](#)). Das bedeutet, wenn Sie GX drücken, wird nach der globalen X-Achse beschränkt, aber beim Drücken von GXX wird nach der X-Achse der Transformationsausrichtung beschränkt.

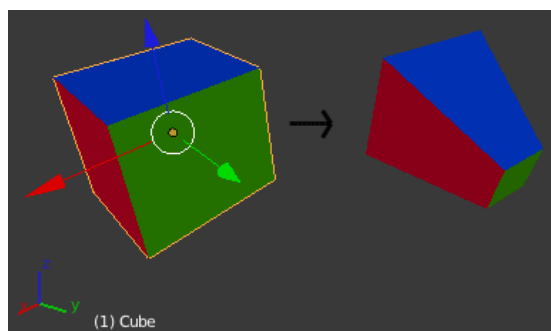


Alt+Leertaste Menü.

Die Optionen zum Ausrichten können in der Kopfzeile der 3D Ansicht festgelegt werden (oder "Fußzeile", da sie sich standardmäßig am unteren Ende der Ansicht befinden), mit AltLeertaste, oder über das Menü Ausrichtung in der Kopfzeile der 3D Ansicht.






Zusätzlich zu den vier eingebauten Optionen, kann eine [benutzerdefinierte Orientierung](#) definiert werden.

Unser Demo Würfel

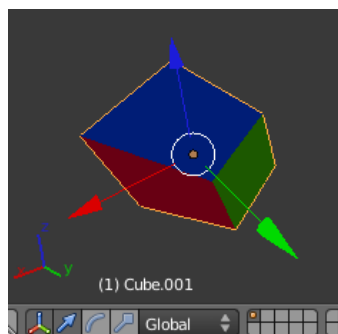


To demonstrate the various behaviors, we add some colors to the default cube, rotate it -15° along its local z- and x-axes, and we scale its "y" face down.

Zwei Dinge gilt es zu beachten:

- Die "Mini-Achse" in der linken unteren Ecke, welche die globale x/y/z Orientierung anzeigt.
- The ["Object Manipulator"](#) widget emanating from the selection, which represents the current Transformationsausrichtung.
 - If you click on one of the axes of the Manipulator with LMB , it will allow you to constrain movement to only this direction. An example of a keyboard equivalent is GZZ.
 - If you  Shift LMB  click, it will lock the axis you clicked on and allow you to move in the plane of the two remaining axes. The keyboard analogue is G ShiftZ.

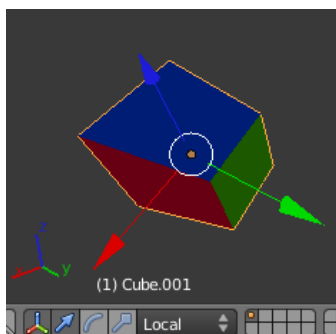
Orientierungen



Global.

Global

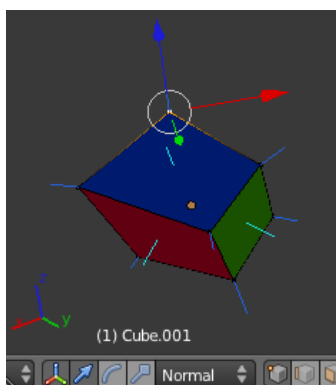
Der Manipulator stimmt mit der globalen Achse überein In this, the Global orientation, the orientation's x,y,z are identical to the world's x,y,z. When this mode is selected an exceptional behavior occurs: GXX uses the Local x-axis instead of the (redundant) Global x-axis.



Local.

Lokal

The manipulator matches the object axis. Notice that, here, the Manipulator is at a slight tilt (it is most visible on the object's y-axis, the green arrow). This is due to our 15° rotation of the object. This demonstrates the difference between local coordinates and global coordinates. If we had rotated the object 90° along its x-axis, we would see that the object's "Up" is the world's "Forward" -- or the object's z-axis would now be the world's y-axis. This orientation has an effect on many parts of the interface, so it is important to understand the distinction.



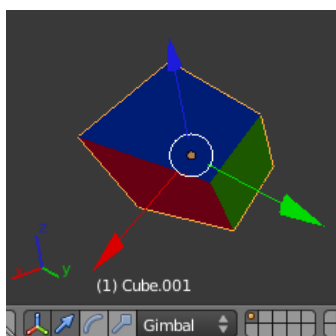
Normal.

Normal

The z-axis of the manipulator will match the normal vector of the selected object. In Object Mode, this is equivalent to Local Orientation, but in Edit Mode, it becomes more interesting.

As you see, the light blue lines indicate the faces' normals, and the darker blue lines indicate the vertex normals (these were turned on in the N Properties Panel under Mesh Display » Normals » Face and Vertex). Selecting any given face will cause our Manipulator's z-axis to align with that normal. The same goes for Vertex Select Mode. Edge Select is different--A selected Edge has the z-axis aligned with it (so you will have to look at the Manipulator widget to determine the direction of x and y). If you select several elements, it will orient towards the average of those normals.

A great example of how this is useful is in Vertex Select Mode: Pick a vertex and then do GZZ to tug it away from the mesh and shove it into the mesh. To make this even more useful, select a nearby vertex and hit ⇧ ShiftR to repeat the same movement—except along that second vertex's normal instead.



Gimbal.

Aufhängung

Gimbal's behavior highly depends on the [Rotation Mode](#) that you are in (accessible in the N Properties Panel in the 3D View, in top section, Transform).

XYZ Euler

the default rotation mode, the object Manipulator's z-axis will always point to the global z-axis, where the other two will remain perpendicular to each other.

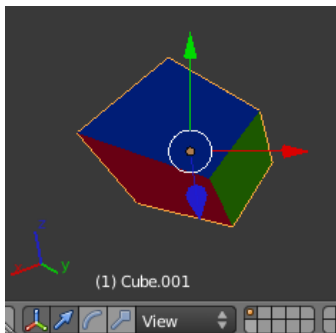
In the other Euler rotation modes, the last axis applied will be the one for which the Manipulator stays fixed. So, for YZX Euler, the x-axis of the Manipulator will be the same as the global x-axis.

Axis Angle

The x, y, and z coordinates define a point relative to the object origin through which an imaginary "skewer" passes. The w value is the rotation of this skewer. Here, the Manipulator's z-axis stays aligned with this skewer.

Quaternion

Though Quaternion rotation is very different from the Euler and Axis Angle rotation modes, the Manipulator behaves the same as in Local mode.



View.

Ansicht

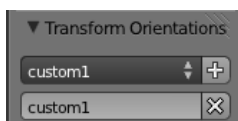
The manipulator will match the 3D view, Y → Up/Down, X → Left/Right, Z → Towards/Away from you.

This way you can constrain movement to one View axis with GXX.

Benutzerdefinierte Orientierungen

Modus: Objektmodus und Editiermodus

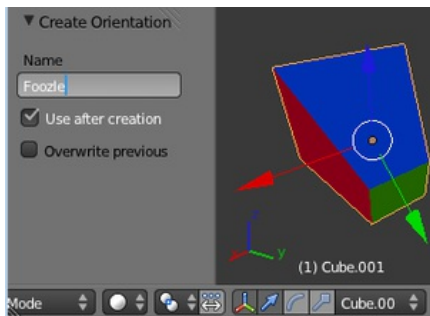
Tastenkombination: CtrlAltLeertaste



Benutzerdefinierte Orientierung

You can define custom Transformationsausrichtungen, using object or mesh elements. Benutzerdefinierte Transformationsausrichtung defined from objects use the local orientation of the object whereas those defined from selected mesh elements (vertices, edges, faces) use the normal orientation of the selection.

The Transformationsausrichtung panel, found in the "N Properties Panel," can be used to manage Transformationsausrichtung: selecting the active orientation, adding and deleting custom orientations.



Renaming a Custom Orientation

The default name for these orientations comes from whatever you have selected. If an edge, it will be titled, "Edge," if an object, it will

take that object's name, etc. The Toolshelf (T in the 3D View) allows you to rename the custom orientation after you hit CtrlAltSpace.

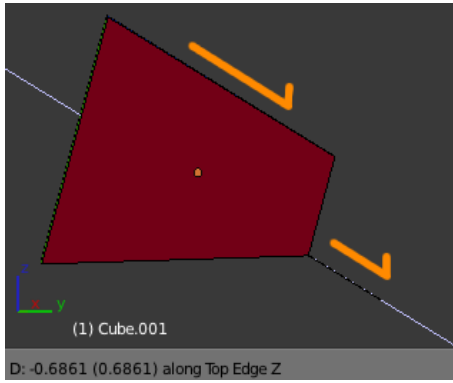



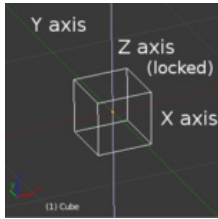
Figure 1.

The technique of creating custom orientations can become important in creating precise meshes. In Figure 1, to achieve this effect:

1. Select the object's sloping top edge
2. Create a Custom Orientation with CtrlAltSpace and rename it "Top Edge".
3. Select the object's bottom, right edge.
4. Extrude with E.
5. Cancel the extrusion's default movement by hitting RMB  or Escape.
6. Hit G to reinitiate movement.
7. Hit ZZ to constrain to the "Top Edge" orientation.

Achsenfestlegung

Beschreibung



Achsenfestlegung

[Grundlegende Transformationen \(Translation/Skalierung/Rotation\)](#) im Objekt und Edit Modus, und Extrusionen im Edit Mode können auf bestimmte Achsen relativ zum [Vergleichskoordinatensystem](#) begrenzt werden.

Benutzung

Eine festgelegte Achse wird in einer helleren Farbe dargestellt als andere. Im Bild rechts zum Beispiel ist die Z-Achse hell blau und die Bewegung ist auf diese Achse begrenzt.

Man kann Achsen auf zwei Arten festlegen:

- Eine Transformationsschnelltaste (G,R,S) und eine Achsentaste (X, Y, Z) drücken, um die Transformation auf die Achse zu begrenzen.
- Eine Transformationsschnelltaste (G,R,S) drücken, die Maus in die gewünschte Richtung bewegen und MMB drücken.

Achsenbegrenzungstypen

Achsenbegrenzung

Mode: Object und Edit Modi (translate, rotate, scale, extrude)

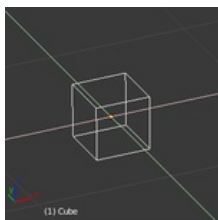
Hotkey: X, Y, Z oder MMB nach dem Bewegen der Maus in die gewünschte Richtung.

Achsenfestlegung begrenzt Transformationen auf eine einzige Achse fest und verhindert Auswirkungen der Transformationen auf andere Achsen. Objekte oder Vertices werden nur auf dieser Achse bewegt.

Ebenenfestlegung

Mode: Object und Edit Modi (translate, scale)

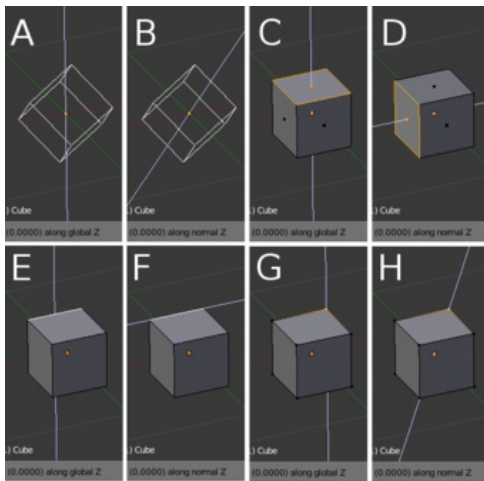
Hotkey: ShiftX, ShiftY, ShiftZ drücken oder die Maus in die gewünschte Richtung bewegen und MMB drücken.



Ebenenfestlegung

Ebenenfestlegung begrenzt Transformationen auf zwei Achsen bzw. eine Ebene.

Achsenbegrenzungs Modi



Achsenbegrenzungs Modi

- x1

Ein einziger Tastendruck begrenzt Bewegungen auf Achsen des globalen Vergleichskoordinatensystems.

- x2

Ein zweiter Tastendruck begrenzt Bewegungen auf Achsen des momentan ausgewählten Vergleichskoordinatensystems.

- x3

Hebt Begrenzungen wieder auf.

Da die Vergleichskoordinatensysteme teilweise kontextabhängig sind, hängen sie von der Auswahl ab wie man im Bild Achsenbegrenzungs Modi Bild sehen kann. A und B zeigen Transformationen im globalen und lokalen Bezugssystem, C und D zeigen dieselbe Situation mit einer Flächenauswahl, E und F mit einer Kantenwahl und G und H mit einer Vertexwahl.

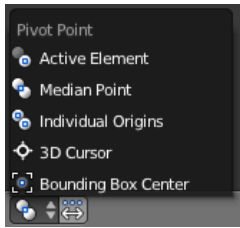
Achsenbegrenzung kann in Kombination mit [Zahleneingaben](#) verwendet werden.

Pivot Punkt

Pivot Punkt Auswahl

Mode: Objekt Modus und Edit Modus

Menu: Droplist im 3D Ansicht Balken



Pivot Punkt Modi

Der Pivot Punkt ist der relative Punkt im Raum von dem aus alle Transformationen ausgehen. Man kann aus fünf allgemeinem Modi für Pivot Punkte aus dem gezeigten Menü auswählen. Es ist nützlich zu wissen welcher der praktischste Modus für eine bestimmte Funktion ist.


- [Active Element](#)
- [Median Punkt](#)
- [Individueller Ursprung](#)
- [3D Cursor](#)
- [Bounding Box Zentrum](#)

Die oben aufgeführten Beispiele benutzen zwar Meshe, würden aber auch mit anderen Objekttypen (Kurven, Oberflächen,etc.) funktionieren.

Aktives Element als Pivot

Mode: Objekt Modus und Edit Modus

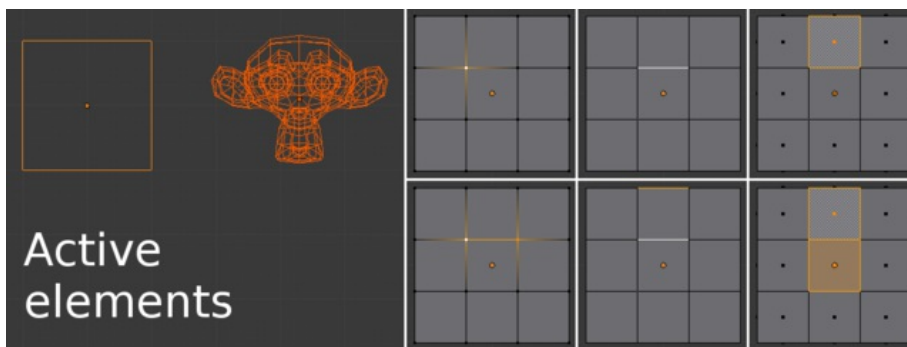
Hotkey: Alt.

Menu: Auswahl vom Knopf im 3D Ansichtsbalken 

Das *aktive* element kann ein Objekt, Vertex, eine Kante oder Fläche sein. Das aktive Element ist das zuletzt ausgewählte und wird in hell organge im Objekt Modus und weiß im Edit Modus dargestellt. Mit aktivem Element als Pivot ausgewählt werden alle Transformationen relativ zum momentan ausgewählten Objekt durchgeführt.

The *active* element can be an Object, vertex, edge or a face. The active element is the last one to be selected and will be shown in a lighter orange color when in Object mode and white when in Edit mode. With Active element as Pivot set to active, all transformations will occur relative to the active element.

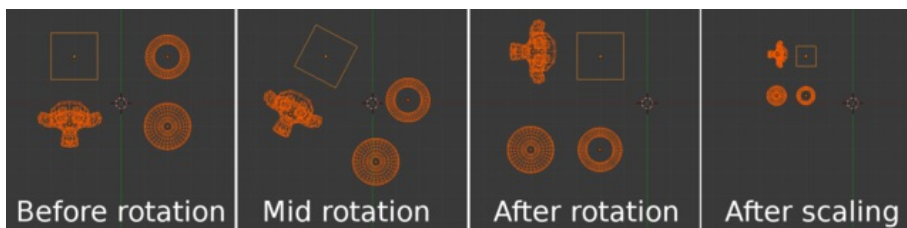
[Lerne mehr über allgemeine Optionen des Pivot Punkts. »](#)



Das darstellen von aktiven Elementen im Objekt Modus wird im linken Bild gezeigt. Aktive Elemente im Edit Modus werden im Bild rechts dargestellt.

Im Objekt Modus

Im Objekt Modus werden Transformationen um das Zentrum des aktiven Objekts durchgeführt. Das zeigt das Bild unten.



Transformation mit einem Würfel als aktivem Element.

Im Edit Modus

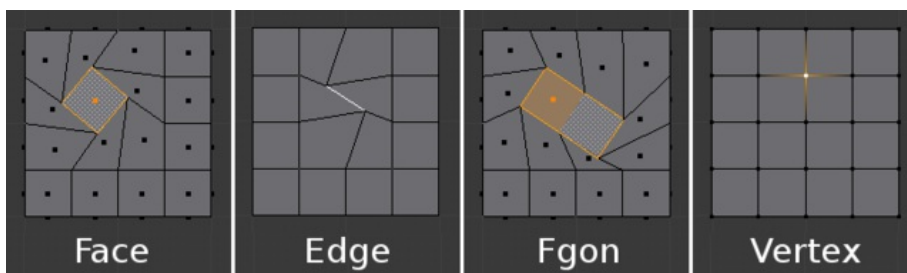
Das aktive Element im Edit Modus hängt von ein paar Regeln ab:

- Der Pivot Punkt ist immer der Median der Koordinaten der aktiven Elemente.
- Transformationen erfolgen durch transformieren die **Vertices** der ausgewählten Elemente.

Hier gibt sind zwei Beispiele:

Einfache Auswahl

Wenn ein einzelnes Element ausgewählt wird, wird es automatisch zum aktiven Element. Im Bild unten kann man sehen das wenn es Transformiert wird, werden seine Vertices bewegt, mit der Konsequenz das alle verbundenen Elemente verformt werden.



Edit Modus und ein einzelnes ausgewähltes Element

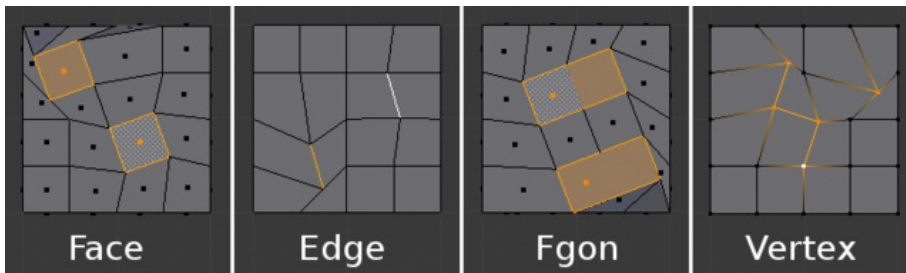
Die verschiedenen Fälle untersucht:

- *Flächen* und *Fgons* haben ihren Pivot Punkt im Median Punkt ihrer Vertices.
- *Kanten* haben ihren Pivot Punkt in der Mitte.

- Ein einzelnes *Vertex* hat keine Dimensionen, also kann es nicht transformiert werden (abgesehen von Translationen, die aber nicht direkt von Pivot Punkten abhängen).

Mehrfache Auswahl

Wenn mehrere Elemente ausgewählt sind, werden alle transformiert. Der Pivot Punkt bleibt am selben Ort wie eben, nur Fgons verhalten sich anders. Im Bild unten wurden die ausgewählten Elemente rotiert.



Edit Modus und mehrfache Auswahl

- Flächen, Kanten, und Fgons tritt gleiches Verhalten auf.
- Bei *Vertices* ist das zuletzt ausgewählte Vertex das aktive Element und alle Transformationen werden relativ zum ihm durchgeführt.

Individuelle Ursprungspunkt als Pivot

Mode: Object mode and Edit mode

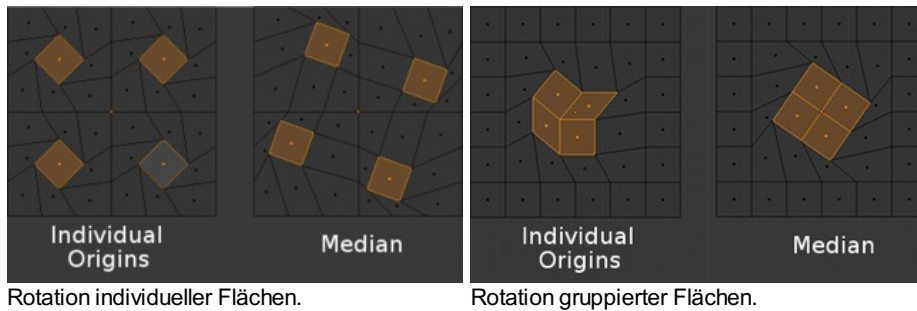
Hotkey: Ctrl.

Menu: Auswahl von diesem Knopf im 3D Ansichtsbalken 

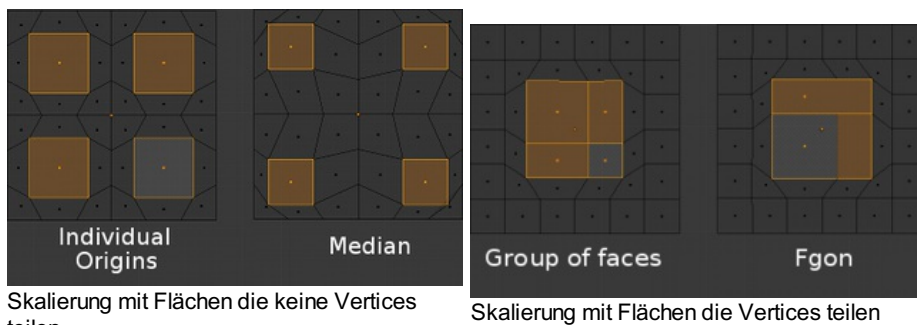
Im Objekt Modus werden die [Objektzentren](#) als Pivot für Transformation von Objekten genommen.

Edit Mode

Im Edit Modus erzeugt die Einstellung individuelle Ursprünge verschiedene Ergebnisse abhängig davon ob der [Auswahlmodus](#) auf Vertex, Kante oder Fläche eingestellt ist. Im Vertex Modus funktioniert es ähnlich wie bei der Einstellung Median und im Kanten Modus werden die Ergebnisse häufig verzerrt. Im Flächen Modus werden die verlässlichsten Ergebnisse erzeugt.



Flächen die Vertices gemein haben werden sich verformen und die bei denen das nicht der Fall ist, rotieren um ihren Median Punkt.



Gruppen von Flächen und Fgons können skaliert werden ohne das ihre äußeren Grenzen verändert werden, allerdings werden die Flächen nicht gleichmäßig skaliert.



Formen mit Flächen und Individuelle Ursprünge als Pivot

Sobald man sich mit den Tücken von [Literall](#) individuellen Ursprüngen vertraut gemacht hat, kann es sehr nützlich sein. Diese Anemone wurde aus einem 12 Seitigen Zylinder in ca 10 Minuten geformt und zwar wiederholtes Extrudieren von individuellen Flächen, Skalierung mit dem [Median als Pivot](#) und Skalierung und Rotation mit individuelle Ursprüngen als Pivot.

3D Cursor als Pivot

Mode: Object mode und Edit mode

Hotkey: .

Der 3D Cursor ist vom Prinzip der einfachste Pivot. Er hat keine besonderen Eigenschaften.

[Lies mehr über Pivot Punkte. »](#)

geometrischer Schwerpunkt as Pivot

Mode: Objekt Modus und Edit Modus

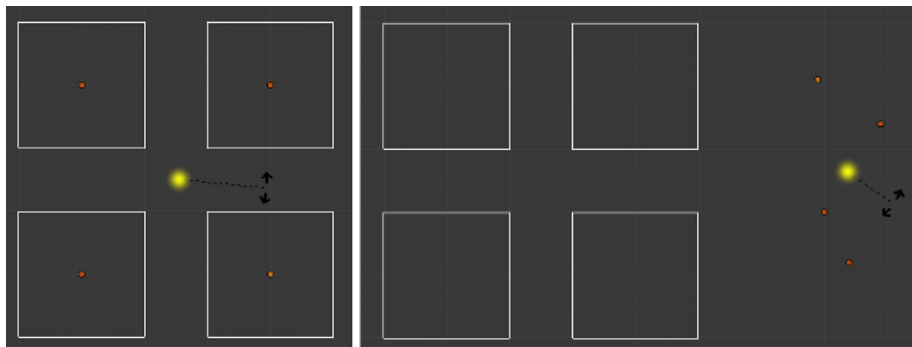
Hotkey: Ctrl,

Menu: Auswahl von diesem Knopf im 3D Ansichtsbalken 

Der Median Punkt ist der Punkt mit den durchschnittlichen Koordinaten aller ausgewählten Punkte.

Im Objekt Modus

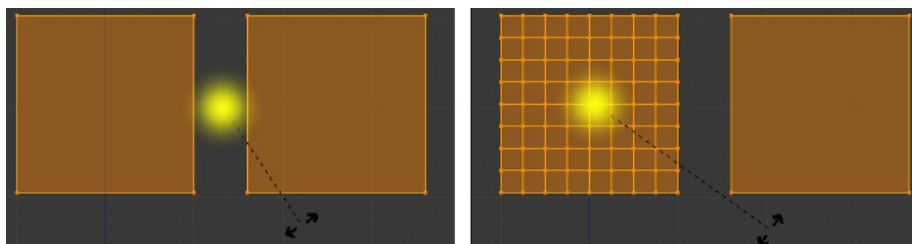
Im Objekt Modus werden die Objektzentren als einzige Punkte betrachtet. Das kann verwirrend sein, denn wie im Bild unten zu sehen ist kann das Objektzentrum ausserhalb des Meshes liegen.



median Punkt im Objekt Modus

Im Edit Modus

Im Edit werden alle Vertices zur Berechnung genommen - je mehr Vertices ein Objekt hat, desto näher ist ihm der median Punkt.




er median Punkt liegt bei dem gelben Punkt.

Bounding Box Center als Pivot

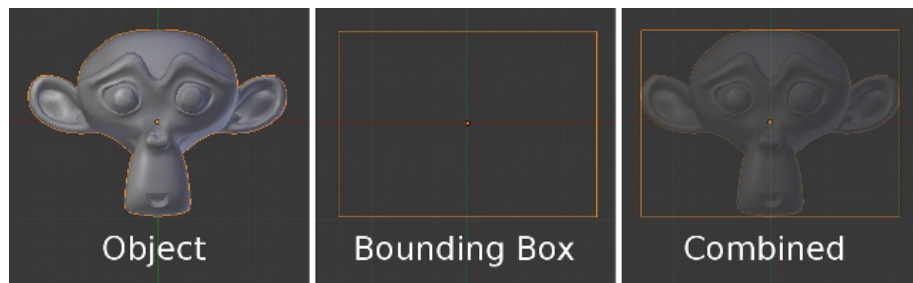
Mode: Objekt Modus und Bearbeitungs Modus

Hotkey: ,

Menu: Wähle  aus dem 3D Ansichtsbalken aus

[Lies hier nach was die Bounding Box ist.](#) Die Bounding Box kann mit der , Taste oder aus dem Pivot Punkt Menü ausgewählt werden. Das Bild unten verdeutlicht was die Bounding Box ist.

[Lies hier mehr über allgemeine oder andere Pivot Punkte »](#)

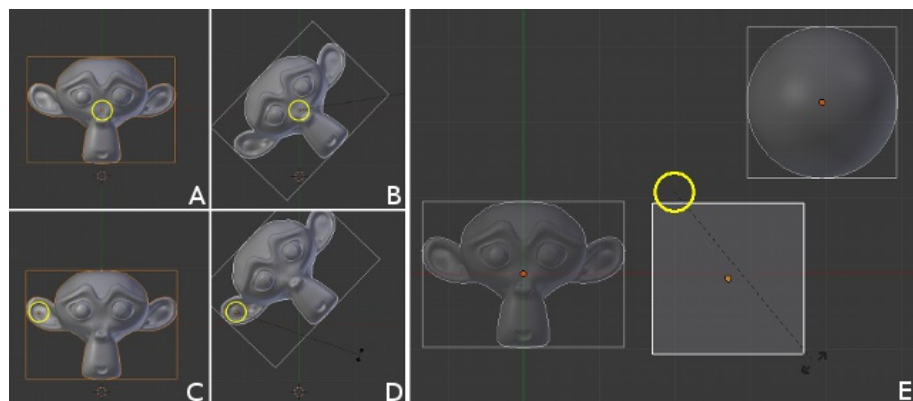


Beziehung zwischen einem Objekt und seiner Bounding Box.

Objekt Modus

Im Objekt Modus hat werden Objekttransformationen relativ zum Zentrum der Bounding Box ausgeführt. Das ist unerheblich und macht keinen Unterschied zum median Punkt wenn nur ein Objekt ausgewählt ist, verhält sich aber sehr anders wenn mehrere Objekte ausgewählt sind.

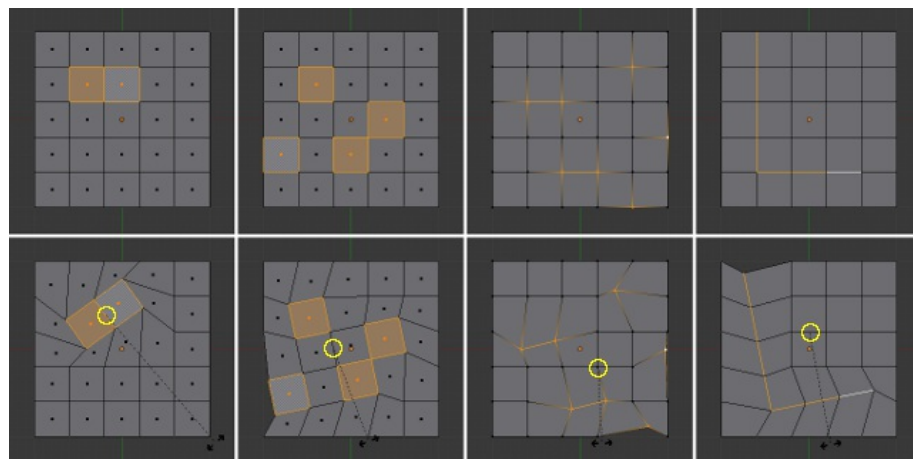
Das wird deutlich im Beispielbild wo sich ein Objekt wie erwartet relativ zum Objektsprung transformiert wird, aber mehrere ihren Pivot Punkt im Zentrum der Bounding Box haben und nicht mehr im relativen median Punkt.



Die vier Bilder links (ABCD) zeigen das Ergebnis einer Rotation mit Bounding Box als Pivot. Das Bild (E) zeigt das Zentrum der Bounding Box bei mehreren ausgewählten Objekten an.

Im Bearbeitungsmodus

Im Bearbeitungsmodus wird der Unterschied zwischen dem median Punkt und der Bounding Box offensichtlich, da der Pivot Punkt nicht mehr von der Dichte der Vertices sondern nur vom Zentrum der Hauptdimensionen abhängt.



Die Auswirkungen der Bounding Box als Pivot im Bearbeitungsmodus. Der Pivot Punkt ist als gelber Kreis gekennzeichnet.

Einrasten

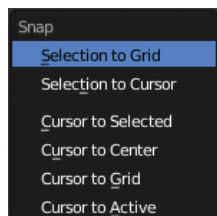
Es gibt zwei verschiedene Einrastoperationen in Blender. Es können Elemente auf bestimmte Punkte eingerastet werden und es können Operationen auf bestimmte Punkte hin einrasten/springen.

Snap

Mode: Objekt Modus und Bearbeitungs Modus.

Hotkey: ⇧ ShiftS

Das Einrastmenü ist verfügbar durch den 3DAnsichtsbalken in Objekt- und Bearbeitungsmodus oder durch das Mesh » Einrasten Menü. Es stellt eine Reihe von Funktionen zur Verfügung um Cursor oder Auswahl auf bestimmte Punkte zu verschieben.



Einrastmenü

Auswahl zum Gitter

Rastet die ausgewählten Objekte auf den nächsten Gitterpunkt.

Auswahl zum Cursor

Rastet die ausgewählten Objekte zur Position des Cursors.

Cursor zur Auswahl

Bewegt den Cursor zum Zentrum der ausgewählten Objekte.

Cursor zum Zentrum

Bewegt den Cursor zum Gitterursprung

Cursor zum Gitter

Bewegt den Cursor zum nächsten Gitterpunkt.

Cursor zum Aktiven

Bewegt den Cursor zum Zentrum des zuletzt ausgewählten Objekts.

Transformationseinrasten

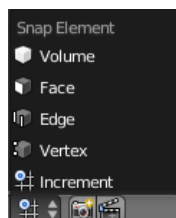
Die Möglichkeit Objekte und Meshelemente auf bestimmte Elemente der Szene einzurasten ist verfügbar durch ein- oder ausschalten des Magneticons im 3D Ansichtsbalken.



Magneticon im 3D Ansichtsbalken bei eingeschaltetem Einrasten.

Einrastmodi

Einrastelement



Einrastelementmenü

Volumen

Rastet die Auswahl auf einen Punkt innerhalb eines Volumens unter dem Mauszeiger ein.

Fläche

Rastet die Auswahl auf einen Punkt innerhalb der Fläche unter dem Mauszeiger ein.

Kante

Rastet die Auswahl auf einen Punkt auf einer Kante unter dem Mauszeiger.

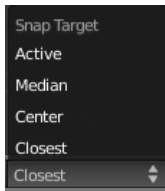
Vertex

Rastet die Auswahl auf einen Vertex in der Nähe des Mauszeigers.

Gitterpunkt

Rastet die Auswahl auf den nächsten Gitterpunkt. In der orthografischen Ansicht hängt die Schrittgröße vom Vergrößerungsfaktor ab.

Einrastziel



Einrastzielenü

Einrastzieloptionen sind verfügbar wenn Vertex, Kante, Fläche oder Volumen als Einrastelement ausgewählt ist.

Aktives Element

Bewegt das aktive Element auf das entsprechende Einrastelement.

Median

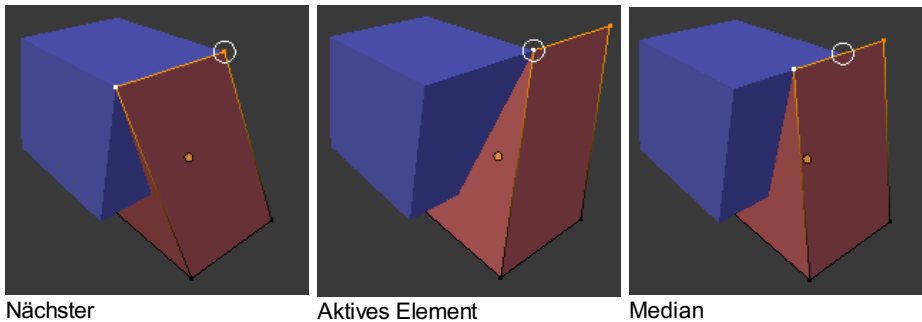
Bewegt den median Punkt der Auswahl auf das entsprechende Einrastelement.

Zentrum

Bewegt das aktuelle Transformations Zentrum zum entsprechenden Einrastelement.

Nächster

Bewegt den zum Einrastelement nächst gelegenen Punkt zum entsprechenden Einrastelement.



Nächster

Aktives Element

Median

Zusätzliche Einrastoptionen



Objektmodus

Bearbeitungsmodus

Wie man an den Rot hervorgehobenen Gebieten im Bild oben sehen kann sind zusätzliche Controller für das Einrastverhalten verfügbar, sie unterscheiden sich in Objekt und Bearbeitungsmodus sowie im Einrastelement. Die vier verfügbaren Optionen sind:

- Die Rotation am Einrastzielausrichten.
- Individuelle Elemente auf die Oberfläche von Objekten zu projizieren
- Elemente auf das eigene Mesh einzurasten.
- Objekte als Ganzes zu betrachten wenn nach dem Objekt Zentrum gesucht wird.

Einrastung von Meshes

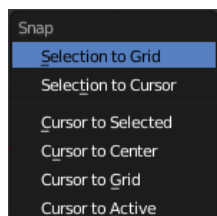
In Blender stehen zwei Typen von Operationen zur Einrastung zur Verfügung. Der erste rastet Ihre Auswahl oder Cursor an einen bestimmten Point ein, während der andere bei Transformationen (Verschiebung, Rotation, Skalierung) genutzt wird, und Ihre Auswahl an Elementen in der Szene einrasten lässt.

Einrasten

Modus: Objektmodus und Editiermodus

Tastenkombination: ⇧ ShiftS

Das Einrasten Menü (auch verfügbar von der Kopfzeile aus der 3D Ansicht im Objektmodus und Editiermodus (Objekt » Einrasten und Mesh » Einrasten). Dieses menü bietet eine Reihe von Optionen an um den Cursor oder Ihre Auswahl zu einem festgelegten Punkt zu bewegen (der Cursor, die Auswahl, oder das Raster).



Snap menu

Auswahl an Raster

Snaps the currently selected object(s) to the nearest grid point.

Auswahl an Cursor

Snaps the currently selected object(s) to the cursor location.

Cursor an Auswahl

Moves the cursor to the center of the selected object(s).

Cursor an Zentrum

Moves the cursor to the center of the grid.

Cursor an Raster

Moves the cursor to the nearest grid point.

Cursor an Aktivem

Moves the cursor to the center of the active object.

Einrastung von Transformierung

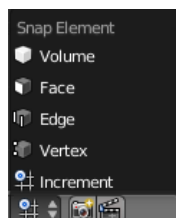
The ability to snap Objects and Mesh element to various types of scene elements during a transformation is available by toggling the magnet icon (which will turn red) in the 3D view's header buttons.



Magnet icon in the 3D view header (red when enabled).

Modi zum Einrasten

Einrasten von Elementen



Snap Element menu

Volume

Snaps to regions within the volume of the first Object found below the mouse cursor. Unlike the other options, this one controls the depth (i.e. Z-coordinates in current view space) of the transformed element. By toggling the button that appears to the right of the snap target menu (see below), target objects will be considered as a whole when determining the volume center.

Face

Snap to the surfaces of faces in mesh objects. Useful for retopologizing.

Edge

Snap to edges of mesh objects.

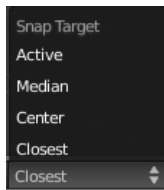
Vertex

Snap to vertices of mesh objects.

Increment

Snap to grid points. When in Orthographic view, the snapping increment changes depending on zoom level.

Einrastziel



Einrastziel
Menü.

Snap target options become active when either Vertex, Edge, Face, or Volume is selected as the snap element. These determine what part of the selection snaps to the target objects.

Aktiv

move the active element (vertex in Edit mode, object in Object mode) to the target.

Mittelwert

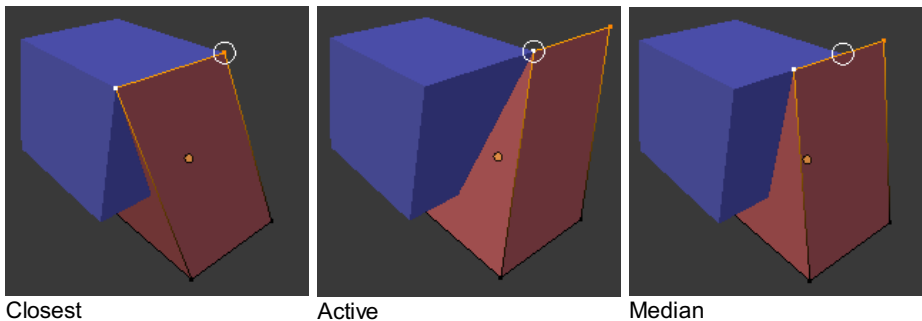
move the median of the selection to the target.

Zentrum

move the current transformation center to the target. Can be used with 3D cursor to snap with an offset.

Entferntester

move the closest point of the selection to the target.



Additional snap options



As seen by the red highlighted areas in the image above, additional controls are available to alter snap behaviour. These options vary between mode (Object and Edit) as well as Snap Element. The four options available are:

- Align rotation with the snapping target.
- Project individual elements on the surface of other objects.
- Snaps elements to its own mesh.
- Consider Objects as whole when finding volume center.

Proportional Bearbeitung

Proportionales Bearbeiten verursacht Veränderungen an nahegelegenen nicht ausgewählten Vertices wenn man Elemente transformiert. Es werden Vertices die näher am ausgewählten und bewegten Vertex liegen stärker bewegt als welche die etwas weiter weg liegen. Es ist deswegen ein sehr nützliches Werkzeug um glatte meshes gleichmäßig und gezielt zu bearbeiten.

Sculpten

Blender hat einen [Sculpt Modus](#) der Werkzeuge zur Verfügung stellt die eine proportionale Bearbeitung ermöglichen ohne das einzelne Vertices angezeigt werden.

Objekt Modus

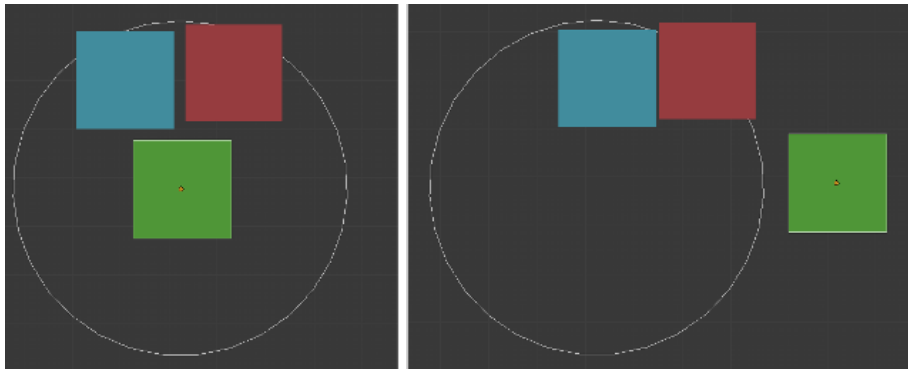
Mode: Objekt Modus

Hotkey: O

Menu: Der Button im 3D Ansichtsbalken



Proportionales Bearbeiten wird normalerweise im Edit mode benutzt, allerdings ist es auch für den Objekt Modus verfügbar. Im Objekt Modus wirkt sich das proportionale Bearbeiten auf ganze Objekte aus anstatt nur auf Meshteile. Im Bild unten ist zu sehen wie die Würfel in blau und rot dem grünen aktiven Würfel folgen wenn dieser bewegt wird und die Würfel in blau und rot innerhalb des Einflussradius liegen.



Proportionales Bearbeiten im Objekt Modus.

Edit Modus

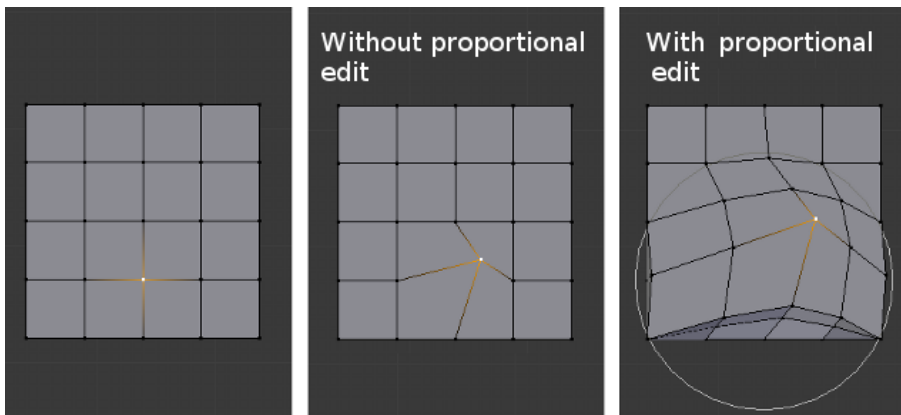
Mode: Edit Modus

Hotkey: O / AltO / ⇧ ShiftO

Menu: Mesh » Proportionales Editing und durch den im Bild unten hervorgehobenen Button



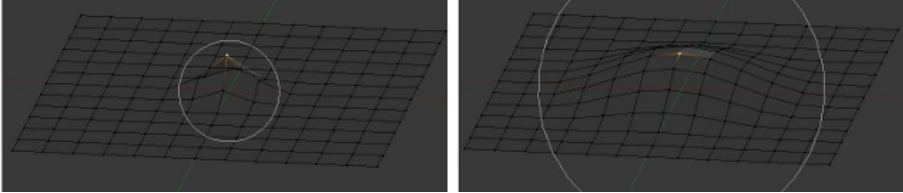
Beim Arbeiten mit dichten Meshes kann es schwer sein geringfügige Veränderungen an diesem Mesh auszuführen ohne sichtbare Beulen oder Falten in der Meshoberfläche zu erzeugen. In solchen Situationen kann das proportionale Bearbeiten sehr hilfreich sein.



Proportionales Bearbeiten im Edit Modus.

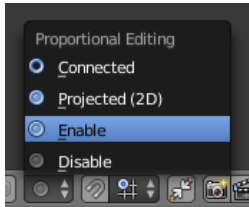
Einfluss

Der Radius des proportionalen Bearbeitens kann mit dem MauseRad WheelUp / WheelDown oder den Tasten PageUp/PageDown eingestellt werden.

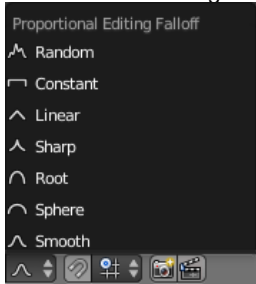


Einflusskreis.

Options



Proportionales Bearbeiten Werkzeug



Falloffmenü

Das Proportionale Bearbeiten Modusmenü ist im 3D Ansichtsbalken verfügbar.

Disable (O oder AltO)

Proportionales Bearbeiten ist ausgeschaltet, nur ausgewählte Vertices werden von Transformationen beeinflusst.

Enable (O oder AltO)

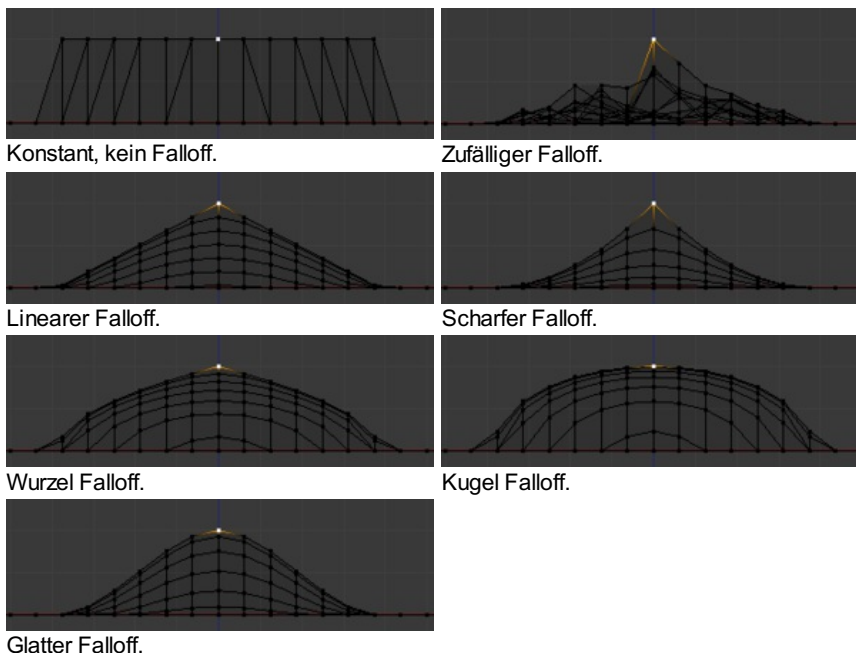
nicht ausgewählte Vertices innerhalb des Radius werden beeinflusst.

Connected (AltO)

Abgesehen vom Radiuseinfluss es ist möglich den proportionalen Falloff durch das Mesh zu verteilen. Vertices die zwar nahe im 3D Raum zum bearbeiteten Vertex sind, aber nicht nahe auf dem Mesh, werden nicht beeinflusst.

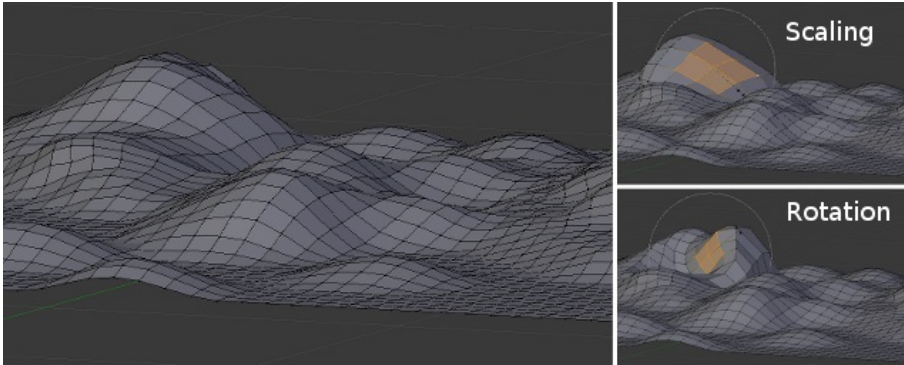
Falloff

Beim Bearbeiten kann das Kurvenprofil das den Falloff bestimmt im Mesh » Proportionaler Falloff Untermenü, dem Werkzeugbalkenicon Falloff menu oder durch wiederholtes drücken von $\text{⇧} \text{O}$ eingestellt werden.



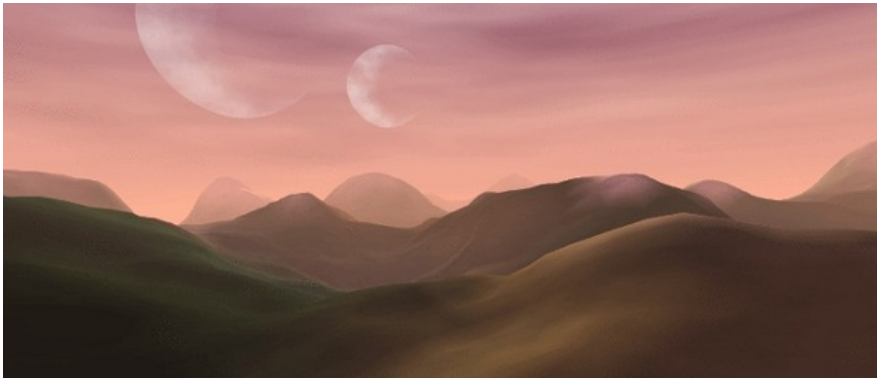
Beispiele

Das proportionale Bearbeiten kann mit den Transformationen großartige Effekte erzielen wie man im Bild unten sehen kann.



Eine Landschaft die durch proportionales Bearbeiten erzeugt wurde.

Kombination dieser Werkzeug mit Vertexbemalung kann phantastische Landschaften erzeugen. Das Bild "fertige Landschaft" unten zeigt wie Ergebnisse des proportionalen Bearbeitens in Verbindung mit Texturen und Beleuchtung aussehen können.



fertige Landschaft

Zeichnen auf der 3D Ansicht

Der Wachsstift ist ein Werkzeug, welches erlaubt Skizzen und Anmerkungen in der 3D Ansicht und in UV/Bild Editor Fenstern mit Freihand Strichen anzufertigen. Dieses Werkzeug kann zurück auf traditionelles 2D Papier und Stift Arbeitsweisen verfolgt werden, wo grobe "Richtlinien" zur Planung und schnellen Kommunikation von Ideen genutzt wurden. In digitalen Bereichen, war eine ähnliche Arbeitsweise präsent, in der mit Wachs Crayons und Stiften auf Monitore gezeichnet wurde. Oft ist es nützlich in der Lage zu sein direkt auf einer Arbeit Skizzen anzufertigen, anstatt einen anderen Ort zu nutzen (z.B. ein anderer Teil des Fensters, oder sogar eine andere Anwendung).

Der Name "Wachsstift" leitet sich von der Hommage an die Wachs Crayons und Stifte ab, die CG Animateur Pioniere nutzten um Bögen und andere Notizen zur Planung auf ihren CRT Bildschirmen zu nutzen.

Zusätzlich zu dem Nutzen für Animature zum Planen von Posen und Bewegungskurven, kann der Wachsstift unter anderem in der folgenden Anzahl von Szenarien von Nutzen sein:

- Planung von Topologie und/oder dem Aussehen von Modellen.
- Director's Foto Review Werkzeug.
- "Whiteboard" und Zuweisungs-Review Werkzeug für Lehrer.



Der Wachsstift in Aktion.

Die nächsten Seiten zeigen wie der Stift genutzt werden kann:

- [Zeichnung von Skizzen.](#)
- [Ebenen und Animation.](#)
- [Konvertierung von Skizzen zu Geometrie.](#)

Drawing With Grease Pencil

1. Enable the Grease Pencil by clicking Draw, Line, Poly or Erase from the Toolshelf (T). A new layer will be automatically added for you to draw on.
2. A new layer can be added from the Grease Pencil Properties panel. This panel can also be used to customise the color, opacity and thickness of the pencil lines. Changes to these settings will affect all strokes on the current layer.



Grease Pencil Tool Shelf and Properties Panel.

Grease Pencil sketches can be converted to editable geometry and used to aid the animation process.

- [Read more about Layers and Animation »](#)
- [Read more about Converting sketches to geometry »](#)

Drawing

The Toolshelf provides a number of options for drawing with the Grease Pencil which are detailed below. The Toolshelf can be seen in the screenshot Grease Pencil Tool Shelf and Properties Panel above.

Grease Pencil Mode and Shortcut Summary

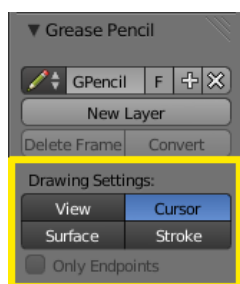
Type	Shortcut	Usage
Draw	D LMB	Draw a new stroke (multiple short, connected lines). The stroke will finish when you release the mouse button.
Line	CtrlD LMB	Draw a new line in rubber band mode. The line will finish when you release the mouse button.
Poly	CtrlD RMB	Draw connected lines by clicking at various points. Lines will be automatically added to connect the two points.
Erase	D RMB	Erases segments of strokes that fall within the radius of the eraser "brush". The erasing will continue until the mouse button is released. If begun with Erase, either RMB or LMB will erase strokes.

Sketching Sessions

A Sketching Session allows for rapid sketching with the Grease Pencil when multiple strokes are desired. With this option set, a sketching session starts when a Grease Pencil stroke is made. The type of session (Draw, Line, Poly, Erase) is determined by the first stroke made which can be done via hotkeys or the Toolshelf. Use Esc or ↵ Enter to exit the sketching session. Note that in a Erase Sketching Session both LMB or RMB can be used once the session has started.

Shared Grease Pencil Settings

Drawing Settings



Grease Pencil Drawing Settings.

In the Grease Pencil Panel of the Properties shelf (N) there are several choices for Drawing Settings.

View

New strokes are locked to the view.

Cursor (3D view only)

New strokes are drawn in 3D-space, with position determined by the 3D cursor and the view rotation at the time of drawing. Cursor is available as an option in the UV/Image Editor but it functions identically to the View option.

Surface (3D view only)

New strokes are drawn in 3D-space, with their position projected onto the first visible surface.

Stroke (3D view only)

New strokes are drawn in 3D-space, with their position projected onto existing visible strokes. Note that strokes created with View are not in 3D-space and are not considered for this projection.

Enabling the Only Endpoints setting applies the drawing setting only to the endpoints of the stroke. The part of the stroke between the endpoints is adjusted to lie on a plane passing through the endpoints.



The effect of different Drawing Settings on Grease Pencil strokes.

Sensitivity When Drawing

The default settings for the sensitivity of mouse/stylus movement when drawing have been set to reduce jitter while still allowing fine movement. However, if these are not appropriate they can be altered in User Preferences window » Editing » Grease Pencil.

Manhattan Distance

The minimum number of pixels the mouse should have moved either horizontally or vertically before the movement is recorded. Decreasing this should work better for curvy lines.

Euclidean Distance

The minimum distance that the mouse should have traveled before movement is recorded.

Eraser Radius

The size of the eraser “brush”.

Smooth Stroke

This turns on the post-processing step of smoothing the stroke to remove jitter. It is only relevant when not drawing straight lines. By default this is enabled. It should be noted that it can often cause “shrinking” of drawings, and may be turned off if the results are not desirable.

Simplify Stroke

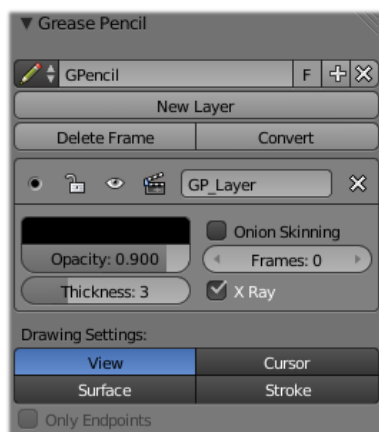
This turns on the post-processing step of simplifying the stroke to remove about half of current points in it. It is only relevant when not drawing straight lines. By default this is disabled. As with Smooth Stroke, it can often cause “shrinking” of drawings, and loss of precision, accuracy and smoothness.

Additional Notes For Tablet Users

- The thickness of a stroke at a particular point is affected by the pressure used when drawing that part of the stroke.
- The “eraser” end of the stylus can be used to erase strokes.

Konvertierung von Strichen zu Objekten

In der 3D Ansicht können Striche der aktiven Ebene zu Geometrie konvertiert werden, basierend auf der aktuellen Ansichtseinstellungen, durch das Transformieren der aufgenommenen Punkte während des Zeichnens (woraus die Striche hervorgehen) in den 3D Raum. Im Moment werden alle Punkte genutzt, also könnte es nötig sein Teile der erstellten Geometrie für den Standardgebrauch zu vereinfachen oder zu unterteilen.



Striche können in Kurven konvertiert werden, wie vom Menü Wachsstift umwandeln hervorgesehen, welches erreichbar ist durch den Knopf Umwandeln in den Wachsstift Einstellungen

- Pfad erstellt NURBS 3D Kurven vom Grad 2 (verhalten sich beispielsweise wie Polylinien).
- Bezierkurve erstellt Bezierkurven, mit frei "zugewiesenen" Anfassern (verhalten sich beispielsweise auch wie Polylinien).

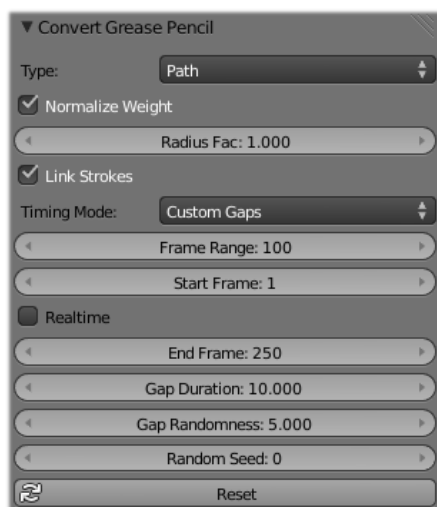
Warum "polylinien-ähnliche" Kurven?

To get by default curves following exactly the grease pencil strokes. If you want a smoothed curve, just edit it to get auto handles (for Bezier), or raise its order (for NURBS).

Konvertierung zu einem Mesh

If you want to convert your sketch to a mesh, simply choose first NURBS, and then convert the created curve to a mesh...

Allgemeine Optionen



Stroke's width will be used to set the curve's control points' radii and weights (**not** NURBS weights, but those used e.g. as goal by the softbody simulation...). The default behavior is to get strokes' width (as defined in its settings – and which might have been modulated by the pen pressure), to multiply it by a given constant (0.1), and to assign it directly to weights. Radii get the same value scaled by the Radius Fac factor (e.g. with a **10.0** factor, a stroke width of **3** will give radii of **3.0**...).

Normalen Gewichtung (standardmäßig aktiv) will scale weights value so that they tightly fit into the `[0.0, 1.0]` range.

All this means that with a pressure tablet, you can directly control the radius and weight of the created curve, which can affect e.g. the width of an extrusion, or the size of an object through a Pfad folgen constraint or Kurven Modifikator!

Pinselfstrichende (enabled by default) will create a single spline (i.e. curve element) from all strokes in active grease pencil layer. This is especially useful if you want to use the curve as a path. All the strokes are linked in the curve by "zero weights/radii" sections.

Timing

Grease pencil now stores "dynamic" data, i.e. how fast they were drawn. When converting to curve, those data can be used to create an Evaluate Time F-Curve (in other words, a path animation), that can be used e.g. to control another object's position along that curve (Pfad folgen constraint, or, through a driver, Kurven Modifikator). So this allows you to reproduce your drawing movements.



All those “timing” options need Link Stroke to be enabled – else they would not make much sense!



Please note that if you use this tool with older grease pencil's strokes (i.e. some without any timing data), you will only have a subset of those options available (namely, only linear progression along the curve over a specified range of frames...).

Begrenzungsmodus

This control let you choose how timing data are used.

No Timing

Just create the curve, without any animation data (hence all following options will be hidden)...

Linear

The path animation will be a linear one.

Original

The path animation will reflect to original timing, including for the “gaps” (i.e. time between strokes drawing).

Benutzerdefiniert

The path animation will reflect to original timing, but the “gaps” will get custom values. This is especially useful if you have very large pauses between some of your strokes, and would rather like to have “reasonable” ones!

Bildbereich

The “length” of the created path animation, in frames. In other words, the highest value of Evaluation Time.

Startbild

The starting frame of the path animation.

Echtzeit

When enabled, the path animation will last exactly the same duration it took you do draw the strokes.

Endbild

When Realtime is disabled, this defines the end frame of the path animation. This means that the drawing timing will be scaled up or down to fit into the specified range.

Dauer

Custom Gaps only. The average duration (in frames) of each gap between actual strokes. Please note that the value entered here will only be exact if Realtime is enabled, else it will be scaled, exactly as the actual strokes' timing is!

Raster-Zufälligkeit

Only when Gap Duration is non-null. The number of frames actual gap duration can vary of. This allows the creation of gaps having an average well defined duration, yet keeping some random variations to avoid an “always the same” effect.

Random Seed

The seed fed to the random generator managing gaps duration variations. Change it to get another set of gaps duration in the path animation.

Beispiel

Hier ist ein Beispiel "handgeschriebenes" Video, dass mit Kurven erstellt wurde, welche aus Zeichen-Daten konvertiert wurden:

[\[video link\]](#)

Hier findet sich die dazugehörige blend Datei : [File:ManGreasePencilConvertToCurveDynamicExample.blend](#)

Modellieren in Blender

Wie Sie im [Quick Start](#) Kapitel gesehen haben, braucht die Entwicklung einer 3D Szene mindestens drei Hauptobjekte: Modelle, Materialien und Lichter. In diesem Teil werden wir uns genauer mit Ersterem beschäftigen: dem Modellieren. Das Modellieren ist die Kunst und Wissenschaft, eine Oberfläche zu schaffen, die die Gestalt eines echten Gegenstands nachahmt oder in Ihre Vorstellungskraft von abstrakten Gegenständen passt.

Objekte haben viele Formen, Gestalten und Größen, so hat Blender viele verschiedene Werkzeuge, die Ihnen helfen, Ihr Modell schnell und effizient zu entwickeln:

[Objekte](#)

Das Arbeiten mit Gegenständen als Ganzes

[Meshes](#)

Das Arbeiten mit dem Mesh (engl.: Masche, Drahtgeflecht, Gittergewebe), das die Gestalt eines Gegenstands definiert

[Kurven](#)

Das Verwenden von Kurven, um Gegenstände zu modellieren und zu kontrollieren

[Oberflächen](#)

Modellieren einer NURBS-Oberfläche

[Text](#)

Textwerkzeuge, um Wörter im 3D Raum zu platzieren

[Meta Objekte](#)

Kleckse und Kugeln

[Duplikationen](#)

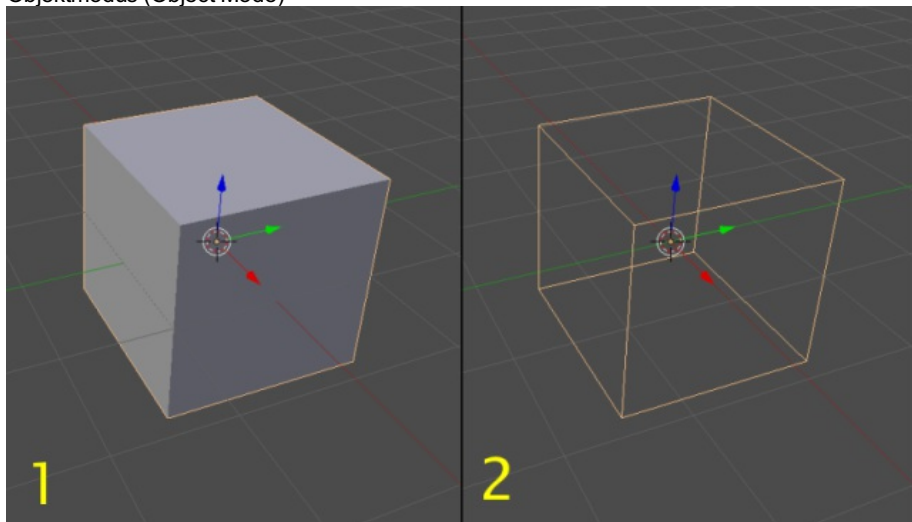
Doppelte Objekte

[Modellier-Skripte](#)

Da die Funktionalität von Blender über Python erweiterbar ist, gibt es mehrere sehr nützliche Skripte, die Ihnen beim Modellieren helfen.

Viele 3D-Modellierer verwenden das "Box-Modelling", das mit einem einfachen Würfel beginnt und durch Extrusion von Flächen und dem Bewegen von Punkten zu größeren und komplexeren Meshes führt. Für flache Objekte, wie Wände und Tischplatten, können Sie das "Curve-Modelling" verwenden, das den Umriss definiert, Bezier oder NURBS-Kurven verwendet, und diese dann zur gewünschten Dicke extrudiert. Jede dieser Methoden wird in Blender vollständig von den Modellierwerkzeugen unterstützt.

Objektmodus (Object Mode)



Ausgewähltes Objekt.

Die Geometrie einer Szene wird von einem oder mehreren Objekten bestimmt, z.B. Lampen, Kurven, Oberflächen, Kameras, Meshes und Grundformen ("Primitives"). Diese werden in "[Mesh Primitives](#)" beschrieben.

Objekttypen

Meshes	Meshes sind Objekte, die aus polygonalen Flächen, Kanten und/oder Eckpunkten bestehen. Sie können in Blender umfassend mit den Mesh-Bearbeitungswerkzeugen zusammengesetzt und editiert werden.
Kurven	Kurven (Curves) sind mathematisch definierte Objekte, die mit handles (engl. f.: Griff, Hebel) oder Kontrollpunkten manipuliert werden können.
Oberflächen	Oberflächen (Surfaces) sind vier- oder mehrseitige Flächen, die über Kontrollpunkten manipuliert werden. Diese sind für organische und abgerundete, aber im Grunde einfache Formen nützlich.
Metabälle	Metabälle (Metas) sind Objekte, die durch eine mathematische Funktion geformt werden. Diese Funktion definiert das 3D-Volumen, in dem das Metaobjekt existiert. Metabälle können tropfenartige Formen schaffen, die flüssigkeitsähnliche Eigenschaften zeigen, sobald zwei oder mehr Metabälle verwendet werden.
Text	Textobjekte (Texts) erzeugen eine 2D-Darstellung einer Zeichenkette.
Armatures	Skelette (Armatures) werden verwendet, um 3D-Modelle so auszurüsten (riggen), dass sie animiert werden können.
Empty	Empties sind leere Objekte, die nicht mit gerendert werden. Sie sind als Knotenobjekte jedoch nützlich, um die Position oder Bewegung anderer Objekte zu kontrollieren.
Kameras	Die Kamera legt den Bildausschnitt fest und speichert diesen. Es können mehrere Kameraobjekte festgelegt werden.
Lampen	Lampen (Lighting) werden verwendet, um Lichtquellen in der Szene zu platzieren.
Kraftfelder	Kraftfelder (Force Fields) simulieren externe Kräfte (Gravitation, Wind, usw.). Sie werden in physikalischen Simulationen verwendet oder um natürliche Bewegungen darzustellen. Im 3D-Editor werden sie durch kleine Kontrollobjekte repräsentiert.



Jedes Objekt kann im Objektmodus bewegt, rotiert und skaliert werden. Jedoch haben nicht alle diese Transformationen auch eine Auswirkung auf die Objekte. Zum Beispiel wird durch Skalieren eines Kraftfeldes dessen Wirkung nicht vergrößert.



Um Änderungen an der Geometrie von editierbaren Objekten vorzunehmen, muss man den Editiermodus verwenden.

Hat man eine Grundform hinzugefügt, bleibt man im Objektmodus. (In früheren Blender-Versionen wurde automatisch in den Editiermodus geschaltet, sobald das Objekt ein Mesh, eine Kurve oder eine Oberfläche war.)

Mit der ⇌ Tab-Taste wechselt man von dem Objektmodus in den Editiermodus. Das Drahtgittermodell der Objektgeometrie sollte jetzt orange dargestellt sein. Dadurch wird angezeigt, dass die Objektgeometrie ausgewählt und aktiv ist. Siehe dazu oben (*Ausgewähltes Objekt*).

Das (*Ausgewähltes Objekt*) Bild zeigt beide Zustände eines Standard-Würfels, nämlich die massive Ansicht und die Drahtgitteransicht. Zum Umschalten zwischen Massiv und Drahtgitter drückt man die Taste Z.

Durch erneutes Drücken der ⇌ Tab-Taste gelangt man wieder zurück in den Objektmodus.

Objektursprungspunkt (Origin oder Pivot)

Jedes Objekt hat einen Ursprungspunkt. Die Position dieses Punktes bestimmt, wo das Objekt im 3D-Raum platziert wird. Wenn ein Objekt ausgewählt wird, erscheint ein kleiner Kreis, der den Ursprungspunkt anzeigt. Die Position des Ursprungspunktes ist wichtig, wenn ein Objekt rotiert oder skaliert wird. Siehe [Pivot Points](#) für weitere Informationen.

Objektursprungspunkt verschieben

Der Ursprungspunkt eines Objektes kann verschoben werden. Dies geschieht im Menü *Object* über das Untermenü *Transform*:

- Geometry to Origin (Geometrie zum Ursprung)

Verschiebt das Modell zum Ursprungspunkt. Der Mittelpunkt des Objekts liegt dann auf dem Ursprungspunkt.

- Origin to Geometry (Ursprung zur Geometrie)

Verschiebt den Ursprungspunkt zum Mittelpunkt des Objektes.

- Origin to 3D Cursor (Ursprung zum 3D-Cursor)

Verschiebt den Ursprungspunkt des Modells an die Position des 3D-Cursors.

- Origin to Center of Mass (Ursprung zum Schwerpunkt)

Verschiebt den Ursprungspunkt auf den Schwerpunkt des Objekts.

Objekte entfernen (Delete)

Modus: Editiermodus oder Objektmodus

Tastenkombination: X oder Del

Menü: Objekt → Löschen...

Entfernen oder Löschen ausgewählter Objekte.

Objekte verbinden (Join)

Modus: Objektmodus

Tastenkombination: StrgJ

Menü: Objekt → Zusammenführen

Verbindet alle ausgewählten Objekte desselben Typs zu einem einzigen Objekt, dessen Ursprungspunkt beim zuletzt aktiven Objekt gesetzt wird. Die Anwendung von *Join* wirkt sich genauso aus wie das Hinzufügen von Objekten im Edit mode. Die nicht-aktiven Objekte werden gelöscht, nur das aktive Objekt bleibt bestehen. Dies ist nur auf editierbare Objekte wie Meshes und Kurven anwendbar.

Einleitung

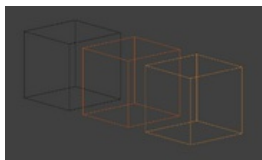
Die Auswahl bestimmt welche Objekte in der Szene verändert werden. Blender hat sowohl im *Objektmodus* als auch im *Editiermodus* fortschrittliche Auswahlmethoden.

Hinweis

In der 3D Ansicht im Objektmodus gibt es ein paar zusätzliche Methoden abgesehen von [den allgemeinen](#). (solange die verlinkte Seite WIP ist gibts hier eine englische WIP ersatzseite.) Diese werden hier aufgeführt.

Auswahl und das aktive Objekt

Blender unterscheidet zwischen zwei verschiedenen Auswahlzuständen:



schwarz: Nicht
ausgewählt; orange:
Ausgewählt; gelb:
Ausgewählt und aktiv

- Im *Objektmodus* ist das zuletzt aus- oder abgewählte Element das "aktive Objekt", dessen Umriss gelb (die der anderen orange) gezeichnet werden. Es gibt zu jeder Zeit genau ein aktives Objekt (auch wenn nichts ausgewählt ist).

Viele Aktionen in Blender verwenden das aktive Objekt als Referenz (z. B. Verknüpfungsoperationen). Wenn Sie bereits eine Auswahl haben und Sie müssen ein anderes Objekt aktiv machen, wählen Sie es einfach neu mit **⇧ Shift RMB** aus.

- Alle anderen ausgewählten Objekte sind lediglich ausgewählt. Sie können eine beliebige Anzahl von Objekten auswählen.

Auswahl mit einem Klick

Sie können ein Objekt mit einem Mausklick auswählen.

Verwenden Sie **RMB** um ein Objekt auszuwählen.

Verwenden Sie **⇧ Shift RMB** um der bestehenden Auswahl ein Objekt hinzuzufügen.

Sollten sich Objekte überlappen erhalten Sie mit **Alt RMB** ein Menü mit einer Auswahl der möglichen Objekte.

Mit **⇧ ShiftAlt RMB** können Sie auf diese Weise einer bestehenden Auswahl ein Objekt hinzufügen.

Verwenden Sie **⇧ Shift RMB** einmal, um ein bereits ausgewähltes Objekt zu aktivieren - und zweimal, falls das Objekt nicht aktiv ist. Dies funktioniert nur, wenn es keine anderen Objekte unter der Maus gibt. Ansonsten werden diese lediglich zur Auswahl hinzugefügt. Es scheint keinen Workaround für diesen Bug zu geben.

Auswahl mit dem Rechteckrahmen

Modus: Objektmodus und Editiermodus

Tastenkombination: B

Menü: Select → Border Select

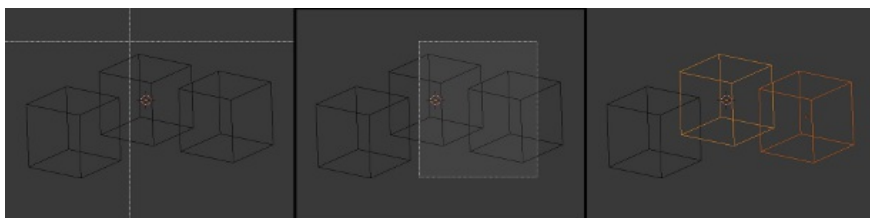
Beschreibung

Bei dieser Methode zeichnen Sie mit gedrückter **LMB** ein Rechteck, um die gewünschten Objekte auszuwählen. Dabei werden auch Objekte ausgewählt, die nur teilweise innerhalb des Rechtecks liegen. Wiederholen Sie den Vorgang, werden noch nicht ausgewählte Objekte der Auswahl hinzugefügt.

Um Objekte abzuwählen, drücken Sie **MMB** oder **⇧ Shift LMB**.

Drücken Sie **RMB**, um den Auswahlvorgang abubrechen.

Beispiel



Auswahl mit einem Rechteckrahmen in drei Schritten

Im ersten Bild wird die aktivierte Rechteckrahmen-Auswahl mit einem gepunkteten Fadenkreuz angezeigt. Im zweiten Bild wird der *Auswahlbereich* durch ein Rechteck mit gedrückter LMB definiert. Innerhalb des Rechtecks befinden sich zwei Rechtecke. Nur diese Rechtecke werden nach Loslassen der LMB ausgewählt.

Beachten Sie im dritten Bild den gelb-umrandeten, mittleren Würfel. Dies ist das aktive Objekt, das zuletzt ausgewählte Objekt vor Verwendung der Rechteckrahmen-Auswahl-Methode.



Tipp

Die Rechteckrahmen-Auswahl-Methode fügt ausgewählte Objekte zur bestehenden Auswahl hinzu. Möchten Sie mit der Auswahl neu beginnen, deselektieren Sie zunächst alle Objekte mit A.

Auswahl mit dem Lasso

Modus: Objektmodus und Editiermodus

Tastenkombination: Ctrl LMB

Menü: Kein Eintrag im Menü

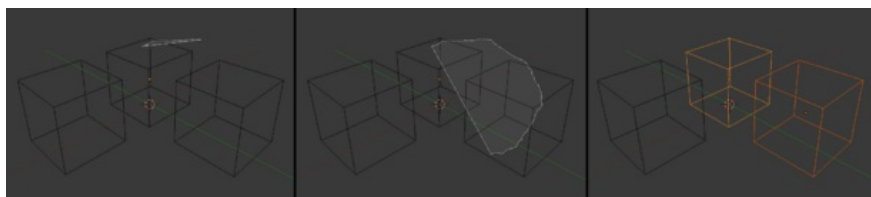
Beschreibung

Die *Lasso-Auswahl*-Methode funktioniert im Objektmodus durch Ziehen einer gepunkteten Linie um die Ursprungspunkte der Objekte.

Verwendung

Halten Sie die Ctrl-Taste gedrückt und umfahren Sie mit gedrückter LMB die Ursprungspunkte derjenigen Objekte, die Sie auswählen wollen.

Wie das *Rechteckrahmen-Auswahl-Werkzeug* fügt das *Lasso* ausgewählte Objekte der Auswahl hinzu. Um Objekte von der Auswahl abzuwählen, drücken Sie Ctrl+ Shift LMB.



Auswahl mit dem Lasso. (Aktives Objekt gelb.)

Auswahl mit dem Kreis

Modus: Objektmodus und Editiermodus

Tastenkombination: C

Menü: Select → Circle Select

Beschreibung

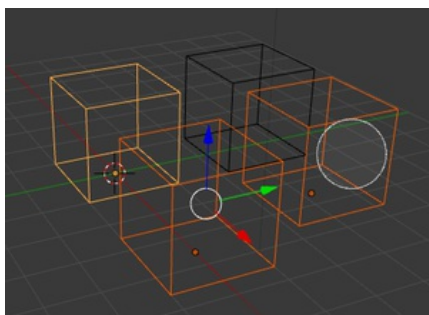


Das Auswahl-Hauptmenü

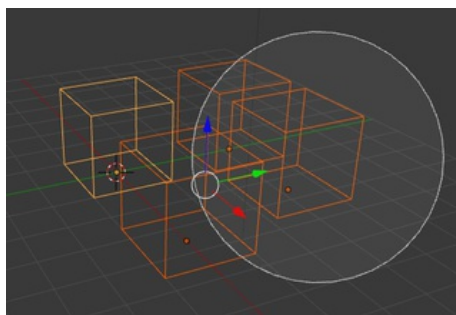
Die Auswahl mit dem Kreis-Methode wird benutzt, in dem Sie mit gedrückter LMB einen gepunkteten Kreis über die Ursprungspunkte derjenigen Objekte bewegen, die Sie auswählen möchten. Jedes Objekt, dessen Ursprungspunkt Sie mit dem *Kreis* berühren, wird ausgewählt. Es ist möglich, dynamisch den Durchmesser des Kreises zu ändern. Betätigen Sie dafür das Mausrad Wheel.

Wie das *Rechteckrahmen-Auswahl-Werkzeug* fügt das *Kreis-Auswahl-Werkzeug* ausgewählte Objekte der Auswahl hinzu. Um Objekte von der Auswahl abzuwählen, drücken Sie Ctrl+ Shift LMB.

Mit RMB oder der Esc-Taste wird der Auswahl-Vorgang beendet.



Auswahl mit dem Kreis mit kleinem ...



... und großem Durchmesser.

Auswahl über Menü

Die oben beschriebenen Selektionsmethoden sind die häufigsten. Es gibt aber auch weitere Auswahloptionen, die über das Select-Menü der 3D-Ansicht erreichbar sind. Sie sind je nach Anwendungsfall sinnvoll einsetzbar.

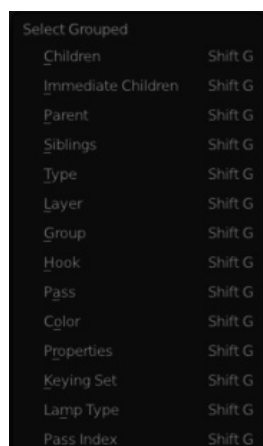
Auswahl gruppierter Objekte

Modus: Objektmodus

Tastenkombination: ⇧ ShiftG

Menü: Auswahl → Gruppe

Beschreibung



Das *selected grouped*-Menü

Es gibt zwei Möglichkeiten, um Objekte in Bezug zueinander anzuordnen. Die erste ist die Eltern-Kind-Beziehung (*parenting*) und die zweite das einfache *Gruppieren*. Wir können diese "Verwandtschaftsverhältnisse" bei der Auswahl von Objekten zu unserem Vorteil nutzen.

Optionen

Auswahl → Gruppe benutzt das aktive Objekt (gelber Umriss) um andere Objekte, die mit dem aktiven Objekt in Verbindung stehen auszuwählen.

Verfügbare Optionen sind:

Children

Wählt alle in der Hierarchie unter dem Objekt stehenden Objekte aus.

Immediate Children

Wählt nur in der Hierarchie eine Ebene unter dem Objekt stehenden Objekte aus.

Parent

Wählt das Übergeordnete Objekt aus.

Siblings (engl. f.: Geschwister)

Wählt alle Objekte aus, die das selbe Elternteil wie das aktive Objekt haben. Dies kann auch verwendet werden, um alle Root-Level-Objekte (Objekte ohne Eltern) zu wählen.

Type

Wählt Objekte, die den gleichen Typ haben wie das aktive Objekt.

Layer

Wählt Objekte, die auf mindestens einer gleichen Ebene (*layer*) liegen.

Group

Wählt Objekte, die Teil einer Gruppe (standardmäßig mit grünem Umriss dargestellt) sind, in der sich das aktive Objekt befindet.

Object Hooks

Jeder Haken am aktiven Objekt.

Pass

Wählt Objekte, die demselben Render-Pass zugeordnet sind. Render-Passes werden in *Properties* → *Object* → *Relations* gesetzt und können im *Node Compositor* (*Add* → *Convertor* → *ID Mask*) verwendet werden.

Color

Wählt Objekte, die dieselbe Objektfarbe (*object color*) wie das aktive Objekt haben. Objektfarben werden in *Properties* → *Object* → *Display* → *Object Color* gesetzt.

Properties

Wählt Objekte, die dieselben *Game Engine Eigenschaften* wie das aktive Objekt haben.

Keying Set

Wählt Objekte, die im aktiven Keying Set enthalten sind.

Lamp Type

Wählt Objekte, die denselben Lampentyp wie das aktive Objekt haben.

Pass Index

Wählt Objekte, die denselben *object pass index* wie das aktive Objekt haben.

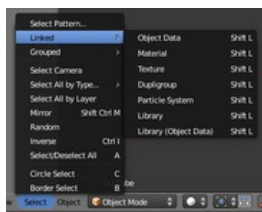
Auswahl über verknüpften Inhalt

Modus: Objektmodus

Tastenkombination: ⇧ ShiftL

Menü: Select → Linked

Beschreibung



Das

Verknüpfungenauswahlmenü

Wählt alle Objekte aus, die einen gemeinsamen Datenblock mit dem aktiven Objekt teilen.

Optionen

Im Objektmodus verwendet *Select* → *Linked* das aktive Objekt als Grundlage zur Auswahl anderer Objekte.

Verfügbare Optionen sind:

Object Data

Wählt alle Objekte aus, die mit den selben Objektdaten verknüpft sind. Zu den Objektdaten gehören die Datenblöcke, die den Typ (Mesh, Kurve, etc.) und den Aufbau des Objektes (konstitutive Elemente wie Eckpunkte, Kontrollpunkte und Angaben über die Position im 3D-Raum) spezifizieren.

Material

Wählt alle Objekte aus, die mit dem selben Material-Datenblock verknüpft sind.

Texture

Wählt alle Objekte aus, die mit dem selben Textur-Datenblock verknüpft sind.

Dupligroup

Wählt alle Objekte aus, die die selbe Gruppe für die Duplikation verwenden.

Particle System

Wählt alle Objekte aus, die das selbe Partikelsystem verwenden.

Library

Wählt alle Objekte aus, die aus der selben [Bibliothek](#) stammen.

Library (Object Data)

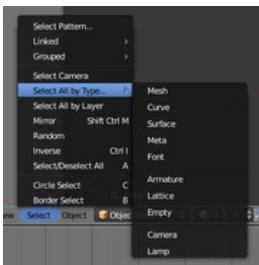
Auswahl nach Typ

Modus: Objektmodus

Tastenkombination: Keiner

Menü: Select → Select All by Type

Beschreibung



Das By Type-Auswahlmenü

Mit dieser Funktion ist es möglich, alle sichtbaren Objekte eines bestimmten Typs auf einmal auszuwählen.

Mögliche Typen sind: Mesh, Curve, Surface, Meta, Font, Armature, Lattice, Empty, Camera, Lamp, Speaker.

Optionen

Im Objektmodus bietet Select All by Type die Möglichkeit zur Auswahl aller Objekttypen, die im *ObData*-Datenblock beschrieben werden können.

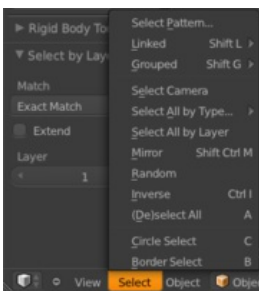
Auswahl nach Layer

Modus: Objektmodus

Tastenkombination: Keine

Menü: Select → Select All by Layer

Beschreibung



Das All by Layer-Auswahlmenü

Ebenen sind ein weiteres Mittel, um Objekte zu gruppieren und zu strukturieren.

Diese Funktion ermöglicht die Auswahl aller Objekte, die sich auf einer angegebenen Layer befinden, egal ob sichtbar oder nicht.

Optionen

Im Tool Shelf → Select by Layer sind die folgenden Optionen verfügbar:

Match

Grad der Übereinstimmung.

- *shared layers*

Die Objekte müssen u. a. auf der unter *layer* angegebenen Layer liegen.

- *exact match*

Die Objekte dürfen nur auf der unter *layer* angegebenen Layer liegen.

Extend

Falls aktiviert, wird die Auswahl erweitert und nicht ersetzt.

Layer

Die Layer auf sich die Objekte befinden müssen.



Tipp: Auswahl von Objekten

Anstatt *Select All by Layer* zu verwenden, kann es effizienter sein, die benötigten Layer sichtbar zu machen und die darauf befindlichen Objekte mit A auszuwählen. Diese Methode ermöglicht es einzelne Objekte von der Auswahl auszuschließen.

Weitere Auswahl-Menüfunktionen

Verfügbare Funktionen auf der ersten Ebene des Menüs sind:

Select Pattern...

Wählt alle Objekte, deren Name einem bestimmten Muster entsprechen. Unterstützte Platzhalter: * passt auf alles, ? entspricht einem einzelnen Zeichen, [abc] entspricht Zeichen in "abc", ![abc] entspricht Zeichen, die nicht in "abc" sind. Es kann zwischen Groß- und Kleinschreibung unterschieden werden.

Beispiel: Haus* passt auf alles das die Zeichenkette "Haus" enthält. Also z. B. "Haustür". Wohingegen Boden* auf alles passt das die Zeichenkette "Boden" enthält. Also z. B. "Bodenbelag".

Select Camera

Selektiert die aktive Kamera.

Mirror (⇧ ShiftCtrlM)

Wählt die Spiegelobjekte des ausgewählten Objekts, z. B. L.sword -> R.sword, aus.

Random

Basierend auf einer prozentualen Wahrscheinlichkeit werden Objekte auf den aktiven Layern zufällig ausgewählt. Nach Betätigung der Funktion wird im Tool Shelf eine Prozentzahl-Eingabebox eingeblendet. Diese Prozentzahl drückt die Wahrscheinlichkeit für jedes einzelnes Objekt aus, ausgewählt zu werden. Es drückt nicht den prozentualen Anteil der ausgewählten Objekte aus.

Inverse (CtrlI)

Selektiert alle Objekte, die nicht ausgewählt waren, während all jene, die ausgewählt waren, deselektiert werden.

(De)select All (A)

Sollte eine Auswahl von Objekten bestehen, werden diese deselektiert. Ansonsten werden alle Objekte selektiert.

Einleitung

In diesem Abschnitt werden Werkzeuge für die Objektbearbeitung im *Objektmodus* beschrieben.

Informationen über einige zusätzliche Möglichkeiten sind in [Bearbeitung in 3D](#) beschrieben.

Objektmodus



Die Objektmodus-Schaltfläche

Standardmäßig werden Blenderdateien im *Objektmodus* geöffnet. Um während der Bearbeitung in den *Objektmodus* zurückzukehren, drücken Sie im 3D View window → Header auf die Objektmodus-Schaltfläche (s. Bild).

Das Bearbeiten von Objekten erfolgt ausschließlich mit selektierten Objekten. Mehr Informationen zum Auswählen von Objekten finden Sie auf [Objektauswahl](#).

Alle hier beschriebenen Befehle finden Sie im *Objekt*-Menü und/oder im *Objekt-Werkzeugkasten*(s. Bilder).



Der Objekt-Menü-Objekt-Werkzeugkasten

Das Erstellen und das Löschen

Zu den grundlegendsten Operationen von Objekten gehört das Erstellen und das Löschen von Objekten. Nachfolgend werden einige Werkzeuge für das Erstellen und das Löschen von Objekten aufgeführt.

Hinzufügen (Add)

Tastenkombination: ⇧ ShiftA

Menü: Main → Add

Folgende Grundobjekte (*primitives*) können hinzugefügt werden:

- **Mesh:** Ebene, Würfel, Kreis, UV-Kugel, Ico-Kugel, Zylinder, Kegel, Raster, Affe ("Suzanne"), Torus.
- **Curve:** Bezier, Kreis, NURBS-Kurve, NURBS-Kreis, Pfad.
- **Surface:** NURBS-Kurve, NURBS-Kreis, NURBS-Fläche, NURBS-Zylinder, NURBS-Kugel, NURBS-Torus.
- **Metaball:** Ball, Kapsel, Ebene, Ellipsoid, Würfel.
- **Text.**
- **Armature:** Einzelner Knochen.
- **Lattice** (engl. f.: Gitter).
- **Empty:** Ebenenachsen, Pfeile, Einzelner Pfeil, Kreis, Würfel, Kugel, Kegel, Bild.
- **Speaker** (engl. f.: Lautsprecher).
- **Camera.**
- **Lamp:** Punkt, Sonne, Spot, Hemisphäre, Fläche.
- **Force Field:** Kraft, Wind, Wirbel, magnetische Kraft, harmonische Kraft, elektrische Ladung, Lennard-Jones, Gewebe, Kurvenführung, Schwarm, Turbulenz, Schleppe, Rauchstrom.
- **Group Instance:** Benutzerdefinierte Objektgruppen.

Vervielfältigung (Duplicate)

Die Vervielfältigungsfunktion gibt es in zwei Varianten.

Tastenkombination: ⇧ ShiftD

Menü: *Object* → *Duplicate Objects*

Diese Variante erstellt eine Kopie des Objekts. Die Kopie ist dabei ein neues Objekt. Spätere Veränderungen an einem Objekt

haben keinen Einfluss auf das andere Objekt.

Tastenkombination: AltD

Menü: *Object* → *Duplicate Linked*

Diese Variante erstellt eine Kopie als Verknüpfung auf das Originalobjekt. Spätere Veränderungen an einem Objekt verändern auch das andere Objekt.

Weitergehende Informationen finden Sie unter: [Vervielfältigung](#).

Verbinden (join)

Tastenkombination: CtrlJ

Menü: *Object* → *Join*

Durch das Verbinden (joining) von Objekten erstellen Sie aus den ausgewählten Objekten ein einzelnes Objekt. Die ausgewählten Objekte müssen dabei vom gleichen Typ und ihre geometrischen Bestandteile editierbar sein (Meshes und Kurven). Das aktive Objekt übernimmt dabei die Geometrie der nicht-aktiven Objekte. Die nicht-aktiven Objekte werden gelöscht. Das aktive Objekt samt dessen Ursprungspunkt bleibt erhalten.



Alternative

Das Ausführen der Verbindungsfunktion hat das gleiche Ergebnis wie das Hinzufügen von neuen Objekten währenddem man das Objekt editiert (Editiermodus).

Löschen (delete)

Tastenkombination: X, D or Delete

Menü: *Object* → *Delete...* → *Delete*

Diese Funktion löscht die ausgewählten Objekte.

Transformations-Werkzeuge

Objekte können in einer Vielzahl von Wegen transformiert werden. Nachfolgend sind verschiedene Arten der Transformation aufgeführt.

Verschieben (Translation)

Tastenkombination: G

Menü: *Object* → *Transform* → *Grab/Move*

Translation bedeutet die Änderung des Standortes von Objekten. Dies ändert die X-, Y- und/oder Z-Koordinaten des Objekt-Ursprungspunkts relativ zum aktuellen Koordinatensprung.

Drehung (Rotation)

Tastenkombination: R

Menü: *Object* → *Transform* → *Rotate*

Die Rotation verändert die Winkel von Objekten. Dies ändert die Rotationswinkel um die X-, Y und / oder Z-Achse des Koordinatensystems des Objekts relativ zum aktuellen Koordinatensystem. Teile innerhalb des gedrehten Objekts verändern dabei ihre Position untereinander nicht.

Größenänderung (Skalierung)

Tastenkombination: S

Menü: *Object* → *Transform* → *Scale*

Die Skalierung verändert die Proportion von Objekten. Die Objekte werden dabei entlang der X-, Y- und/oder Z-Achse gestreckt bzw. gestaucht.

Gruppen und Hierarchien (Parenting)

Da es gleichzeitig sehr viele Objekte in einer Szene geben kann, gibt es in Blender eine Gruppierungsfunktion mit der ausgewählte, z.B. zusammenhängende, Objekte in Gruppen zusammengefasst werden können oder es gibt Fälle wo zur Strukturierung klare Hierarchien benötigt werden. Hierfür bietet Blender die Möglichkeit, Objekte in eine "Eltern-Kind-Beziehung" zu setzen (parenting). Das Verschieben, Rotieren oder die Größenänderung des übergeordneten Objekts (parent) verändert dann auch die untergeordneten Objekte (children).

Objekthierarchien (Parenting)



Das Set-Parent-To Pop-up-Menü

Um ein Objekt einem anderen unterzuordnen müssen mindestens zwei Objekte ausgewählt werden. Das oder die zuerst ausgewählten Objekte (children) werden mit StrgP unter das zuletzt ausgewählte Objekt (parent) eingeordnet. Im *Set-Parent-To* Pop-up-Menü haben Sie die Möglichkeit, eine von mehreren Hierarchiearten auszuwählen. Nach Auswahl wird die entsprechende Beziehung zwischen den Objekten hergestellt.

Das hat zur Folge das die vorher absolute Position der untergeordneten Objekte nun relativ zum übergeordneten Objekt ist.

Mit \diamond ShiftCtrlP kann eine alternative Unterordnung ausgeführt werden. Im normalen Unterordnen werden die globalen Koordinaten des unterzuordnenden Objekts zuerst in lokale Koordinaten bezüglich des in Zukünftigen Übergeordneten Objekts umgerechnet. Dieser umrechnungsschritt findet bei \diamond ShiftCtrlP nicht statt, es wird einfach der Ursprung des Unterzuordnenden Objekts auf den des Übergeordneten Objekts gelegt.

Vertex Unterordnung (Parent)

Es können Objekte einzelnen oder einer Gruppe von 3 Vertices untergeordnet werden sodass die Position der Objekte in Zukunft relativ zu den Vertices ist. Dazu wird erst das/die unterzuordnenden Objekte ausgewählt und dann das Objekt das die gewünschten Vertices enthält. In den Editmodus gewechselt und nach auswählen der gewünschten Vertices löst StrgP das Unterordnen aus.

Note

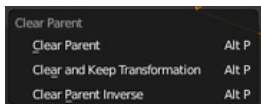
Das ist eine Art umgekehrter [Haken](#)

Optionen

Untergeordnete Objekte Bewegen

Ein untergeordnetes Objekt kann zum übergeordneten durch drücken von AltO bewegt werden. Die Beziehung der Objekte wird dabei nicht beeinflusst.

Beziehung/Unterordnung entfernen



Parent entfernen pop-up menü

Eine solche Beziehung kann mit AltP aufgehoben werden.

Das Menü enthält:

Löse Unterordnung

Falls eine Unterordnung vorhanden ist wird die Verbindung aufgelöst und das Untergeordnete Objekt wird auf die globale Position zurückgesetzt.

Löse Unterordnung, behalte Transformation

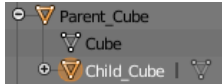
Löst die Verbindung aber rechnet die lokale Position des zu lösenden Objekts so um das es nicht verschoben wird wenn es in globale Koordinaten versetzt wird.

Löse Unterordnung invers

Verschiebt die Untergeordneten Objekte beim lösen der Verbindung als ob die Position des Übergeordneten Objekts von der neuen globalen Position abgezogen werden müsste.

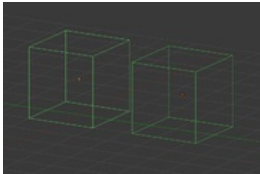
Hinweis

Beachte zum Unterordnen von Objekten auch die



Outliner Ansicht

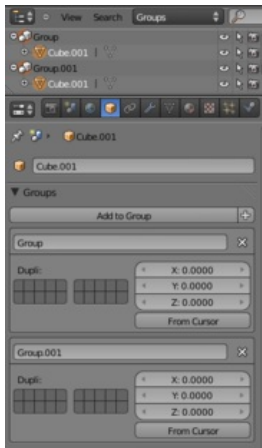
Objekte Gruppieren



Gruppierte Objekte

Gruppierte Objekte haben keine Transformationsbeziehungen untereinander. Sie werden nur zu organisatorischen Zwecken zusammengefasst.

CtrlG erzeugt eine Gruppe aus allen ausgewählten Objekten.



Eine Gruppe benennen

Der Name der Gruppe ist in der Beziehungen Sektion im Objekt Eigenschaften Panel zu finden.

Gruppen können im Outliner durch öffnen des Kontextmenüs umbenannt werden.

Der Gruppeninhalt von Gruppeninstanzen kann durch Begrenzungen durch Ebenen eingeschränkt werden. In den Gruppeneinstellungen können die Ebenen eingestellt werden die von [Dupligruppen](#) dargestellt werden.

Gruppen aus anderen .blend Dateien können eingefügt oder in die aktuelle Datei [über Bibliotheken verknüpft](#) werden.

Duplikation

Mode: Edit- und Objektmodus modes

Hotkey: ⇧ ShiftD

Menu: Object → Duplicate

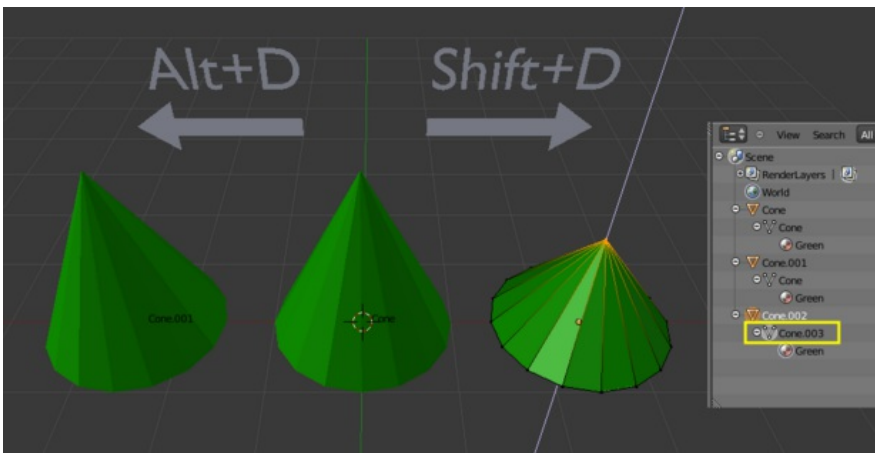
Beschreibung

Duplikation erzeugt eine Kopie der ausgewählten Objekte. Die Kopie wird zwar am selben Ort wie das Original erzeugt, aber der Befehl geht sofort in die [Bewegen](#) Funktion über.

Die Kopie ist ein neues, eigenständiges Objekt, es teilt sich dementsprechend einige Datenblöcke (wie Materialien und IPOs) und hat andere kopiert (wie das Mesh). Welche Datenblöcke nur geteilt und welche tatsächlich kopiert werden sollen kann in den [Benutzer Einstellungen](#) im Datei menü im Bearbeitungs Tab eingestellt werden.

Siehe [Datenblöcke](#) für mehr Informationen wie sie funktionieren.

Bispiele



Das Mesh Cone.006 des Objekts Cone.002 ist bearbeitet worden. Der Unique datablock ID name des Meshes ist im [Outliner](#) hervorgehoben.

Der Kegel in der Mitte wurde (1) verknüpft nach links dupliziert und (2) nach rechts dupliziert.

- Der rechte Kegel kann bearbeitet werden, während der ursprüngliche Kegel unverändert bleibt. Die Meshdaten wurden nicht verknüpft.

Verknüpftes Duplizieren

Mode: Objektmodus

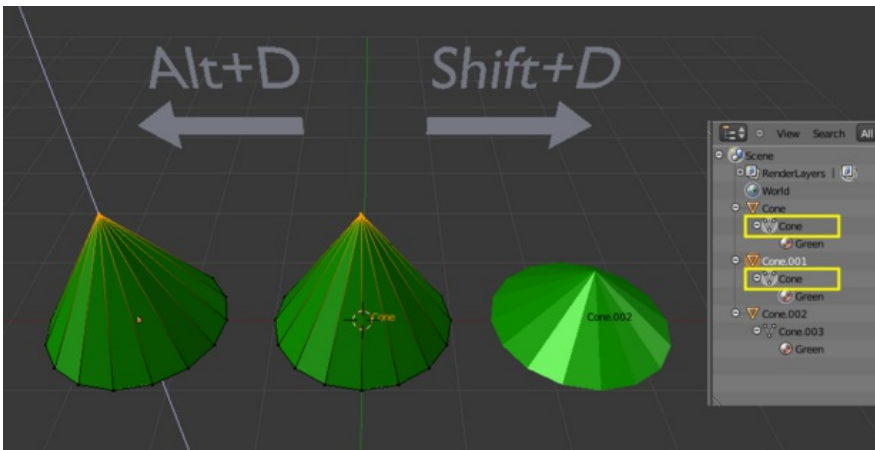
Hotkey: AltD

Menu: Objekt → Duplicate Linked

Beschreibung

Ein verknüpftes Duplikat teilt alle Daten des alten Objektes, es hat keine eigenen Eigenschaften. Das bedeutet wenn eine Eigenschaft an einem der verknüpften Objekte verändert wird, teilen alle anderen diese Veränderung.

Beispiel



Der Cone.001 wurde verknüpft dupliziert. Obwohl es verschiedene Objekte sind haben beide nur ein Mesh. Es ist im [Outliner](#) hervorgehoben.

Das kann sehr nützlich sein bei wiederholt auftretenden Modellen in der Szene und erlaubt ein platzieren bevor die Objekte fertig sind. So kann ein Ursprungsmodell erstellt und verknüpft dupliziert werden, sodass der Gesamteindruck bereits da ist und durch die verknüpfte Veränderung eines Objekts verfeinert werden kann.

Andere Duplikationsmethoden

Mode: Object mode and Edit mode

Panel: Object settings

Es sind im Moment vier Methoden verfügbar um Objekte auf eine Weise zu duplizieren die sonst sehr aufwändig wäre.

[Verts](#)

Diese Funktion erzeugt Duplikate aller untergeordneten Objekte auf jedem Vertex des aktuellen Meshes.

[Faces](#)

Diese Funktion erzeugt Duplikate aller untergeordneten Objekte auf jeder Fläche des aktuellen Meshes.

[Group](#)

Diese Funktion kopiert die Gruppeninstanzierung des zuletzt ausgewählten Objekts.

[Frames](#)

Diese Funktion erzeugt Duplikate des Objekts in jedem Frame. Das kann eine sehr nützliche Funktion zum positionieren oder modellieren sein.

Verknüpfte Bibliotheks Duplikation

Hotkey: ⇧ ShiftF1

Menu: File → Link Append

[Linked Libraries](#)

Bibliotheken sind auch eine Form der Duplikation. Jedes Objekt aus irgendeiner anderen .blend datei kann in die aktuellen verknüpft oder kopiert werden.

Hinweise

- Siehe [Parenting](#) für Duplikation von Transformationseigenschaften.

Vertex Duplikation

Mode: Objektmodus

Panel: Objekt → Duplikation

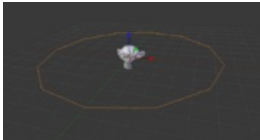
Vertex Duplikation ist die Duplizierung eines Basisobjekts an den Positionen der Vertices eines Meshs. In anderen Worten werden Instanzen des Basis Objekts auf jedem Vertex plaziert.

Download Beispiel .blend Datei

Hier kann eine Beispieldatei heruntergeladen werden. In [dieser Datei](#) ist auf Ebene 1 ein Affe einem Kreis und auf Ebene 2 einer Ikkugel. Die Datei wurde mit blender 2.55 erstellt.

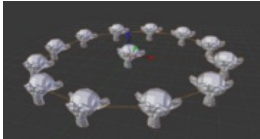
Benutzung

Einrichtung



Ein Affenkopf und ein Kreis

Es werden zwei Objekte benötigt: eines das vervielfältigt werden soll und eins das als Muster dient *wie* vervielfältigt wird. Hier wird als Ursprungsmodell ein Affenkopf und ein Kreis als Muster benutzt.



Vertexduplizierte Affenköpfe

Der Affenkopf muss dem Kreis untergeordnet werden, siehe dafür [Parenting](#). Im Objekt Eigenschaften Editor in der Objektteilung muss nun unter Duplizierung von *Keine* auf *Vertex* umgeschaltet werden. Es können nun beide Objekte nach Wunsch verändert werden.

Ausrichtung



Ausrichtung eingeschaltet, Orientierung +Y

Die Ausrichtung der duplizierten Objekte kann durch die Normalen der Vertices des Musters beeinflusst werden. Hierfür muss Rotation im Duplikationspanel eingeschaltet werden. Um die Ausrichtung aufgrund der Vertexnormalen zu verändern muss in den Beziehungsextras des vervielfältigten Basisobjekts die "Tracking Axis" umgestellt werden.

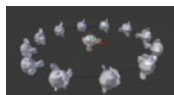
Auswirkung der verschiedenen Ausrichtungen



Negative Y



Positive X



Positive Z, X oben

Andere Duplikationsmethoden werden [hier](#) aufgeführt.

Flächen Duplikation

Mode: Objektmodus

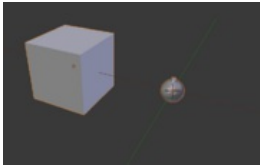
Panel: Objekt → Duplizierung

Flächen Duplikation oder DupliFlächen erlauben es ein Objekt auf den Flächen eines Übergeordneten Objekts zu duplizieren.

Beispiel .blend Datei

Lade [hier](#) die Beispieldatei für diese Seite herunter

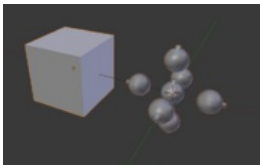
Benutzung



Ein Würfel und eine Kugel

In diesem Beispiel wird eine Kugel mit einem "Nordpol" als Basis und ein Würfel als übergeordnetes Objekt benutzt.

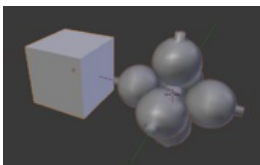
Dafür [Parenting](#).



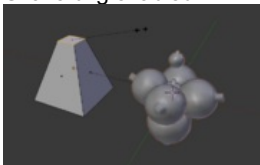
Flächen Duplikation
angewendet auf den
Würfel

Danach muss im Objekt Eigenschaften Editor in den Objekt Eigenschaften im Duplikations Panel der Modus von "Keiner" auf "Flächen" umgestellt werden. Die Kugel wird für jede Fläche auf dem Würfel dupliziert und nimmt dort die Ausrichtung der Flächennormalen an. Beide Objekte werden zwar in der 3D Ansicht noch dargestellt, aber keins wird gerendert.

Skalierung



Skalierung enabled



herunterskalierte obere
Fläche des Würfels

Durch einschalten der Skalierung wird die gröÙe des duplizierten Objekts von der gröÙe der Fläche abhängig gemacht auf die es dupliziert wird.

Gruppen Duplikation

Mode: Objektmodus

Panel: Objekt → Duplikation → Gruppe

Gruppen Duplikation erzeugt Instanzen von Gruppenobjekten

Benutzung

Gruppen Duplikation über die Objekt Eigenschaften im Duplikations Panel funktioniert wie das hinzufügen von Gruppeninstanzen, nur per Hand. [Lies hier nach wie man Objektgruppen erstellt.](#)

Es wird ein Kontrollobjekt und eine Gruppe benötigt. Im Duplikationspanel muss der Duplikationsmodus von "Kein" auf "Gruppe" umgestellt und die gewünschte Gruppe aus dem Menü das sich öffnet ausgewählt werden.

Gruppen Duplikation und dynamisches Verknüpfen

Siehe [Ahänge und Verknüpfungen](#) um zu lernen wie man aus anderen .blend Dateien Objekte verknüpft oder anhängt. Beim verknüpfen von Gruppen wird die Gruppe nicht angezeigt bis nicht eine Gruppeninstanz erzeugt wird.

Echte Kopien erzeugen

Es kann sein das es nötig wird das Objekt zu verändern. Bei verknüpften Objekten ist das nicht möglich ohne die Ursprungsdatei und damit jede davon abhängende zu verändern. Deswegen können lokale Kopien erzeugt werden, die dann mehr Speicherplatz brauchen, aber unbegrenzt verändert werden können. Dazu muss nur Strg+ ShiftA bei ausgewählter Gruppe gedrückt werden um eine lokale Kopie zu erzeugen.

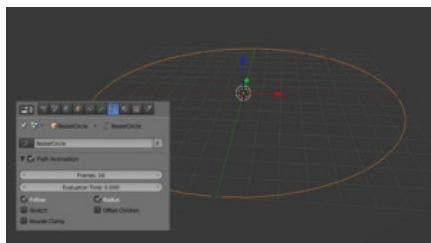
Einzelbild Duplikation

Einzelbild Duplikation erzeugt Objekte entlang eines Pfades.

Benutzung

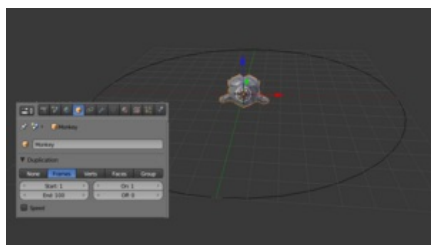
Das gewünschte Objekt muss einer [Kurve untergeordnet](#) werden. In den Kurveinstellungen im "Animationspfad" gibt die Anzahl der Einzelbilder an, in wieviele Teilschritte die Kurve zerlegt wird. Außerdem muss im Panel "Folgen" eingeschaltet sein. Im Objekt muss die Duplikation von "Keine" auf "Einzelbild" umgestellt werden. Es werden nun entlang des Pfades für die angegebenen Einzelbilder (zwischen "Start" und "Ende") nach dem angegebenen Muster (z.B. einen Schritt ein Objekt den nächsten keins) Kopien des Objekts auf den für den Pfad eingestellten Einzelbildern erzeugt und linear platziert. Es funktioniert also ähnlich wie eine Kombination aus [Array](#) mit einem [Kurven Modifikator](#).

Beispiel



Settings for the curve

Es wird eine Kurve hinzugefügt, "Follow" wird eingeschaltet und die Einzelbildanzahl der Kurve auf 16 gesetzt.

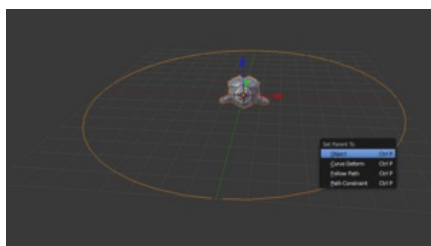


Settings for the object

Es wird ein Objekt hinzugefügt, Einzelbildduplikation wird eingeschaltet und "Geschwindigkeit" wird ausgeschaltet.

Geschwindigkeit

Die Geschwindigkeit Option wird in der Animation verwendet.



Unterordnen



Ausrichtungsveränderung

Die Ausrichtung des Objekts kann ganz normal im Bearbeitungsmodus oder aber auch in den Tracking Axes eingestellt werden.

Mit **Ctrl+Shift+A** werden alle Objekte in echte Kopien umgewandelt.

Hinweis

Es gibt viele Alternative zu dieser Vorgehensweise und es hängt vom Kontext ab, welche die beste ist.

Externe Links

- [Blender Artists: *Dupliframes in 2.5*](#)

Bearbeitungsmodus

Benutzen des Bearbeitungsmodus

Sie können auf zwei Weisen mit geometrischen Objekten arbeiten.

Objektmodus

Operationen im Objektmodus betreffen das ganze Objekt.

Der Objektmodus hat die folgende Kopfzeile in der 3D Ansicht:



Objektmodus Kopfzeile

Bearbeitungsmodus

Operationen im Bearbeitungsmodus beeinflussen nur die Geometrie eines Objektes, aber nicht globale Eigenschaften wie die Position oder Rotation.

Der Bearbeitungsmodus hat die folgende Kopfzeile in der 3D Ansicht:



Bearbeitungs Kopfzeile

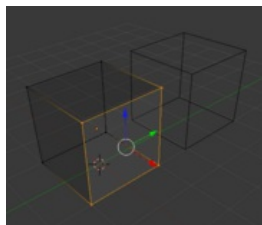
Werkzeuge und Modi in der Kopfzeile der 3D Ansicht sind (von links nach rechts):

Ansicht, Auswählen, Objekt Menüs
Blender Modus
Anzeigemethode für die 3D Ansicht
Pivot Punkt
3D Manipulator Widget
Auswahlmodus
Tiefenpuffer Clipping (hide)
Proportionales Bearbeiten
Einrasten
OpenGL Render

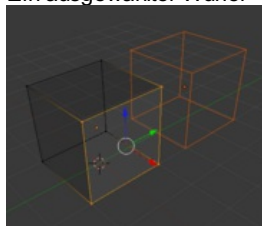
Sie wechseln zwischen den beiden Modi mit der Taste \leftrightarrow Tab oder wählen den gewünschten Modus im Menü der 3D Ansicht.

Nach der Erstellung eines Objektes können Sie immer sofort automatisch in den Bearbeitungsmodus wechseln, wenn Sie die Schaltfläche zum Bearbeitungsmodus wechseln in den [Benutzereinstellungen](#) Editing aktiviert haben. Der Bearbeitungsmodus bezieht sich immer nur das "aktive" bzw. das zuletzt ausgewählte Objekt.

Visualisierung



Ein ausgewählter Würfel



Zwei ausgewählte Würfel
bevor in den
Bearbeitungsmodus
gewechselt wurde

Standardmäßig hebt Blender die ausgewählte Geometrie in orange im Objektmodus und Editiermodus hervor. Die Farbe kann in den [Benutzereinstellungen](#) (StrlAltU → Themen) geändert werden.

Im Objektmodus mit aktiviertem Drahtgittermodell(Z), werden Objekte schwarz angezeigt wenn sie nicht ausgewählt und in orange wenn sie ausgewählt sind. Wenn mehr als ein Objekt ausgewählt ist, werden alle ausgewählten Objekte außer dem aktiven Objekt, dem zuletzt ausgewählten, in einem dunkleren Orange angezeigt. Ganz ähnlich werden im Bearbeitungsmodus, nicht ausgewählte Geometrien in Schwarz dargestellt.

Im Bearbeitungsmodus, kann immer nur ein Objekt editiert werden. Sie können jedoch im Objektmodus) verschiedene markierte Objekte mit StrgJ zu einem Objekt zusammenfassen und dieses dann im Bearbeitungsmodus wiederum mit P zerlegen. Haben Sie verschiedene Objekte ausgewählt, bevor Sie in den Bearbeitungsmodus wechseln, bleiben alle Objekte orange hervorgehoben, um anzuzeigen, dass sie alle Teil einer Auswahl sind.

Sind zwei selektierte Knotenpunkte(vertices) durch eine Kante (edge) verbunden Vertex Auswahlmodus, wird die Kante zwischen den Punkten ebenfalls hervorgehoben. Ebenso wird eine Fläche (face) hervorgehoben, wenn sie durch genügend angrenzende und selektierte Vertices oder Edges eindeutig definiert ist.

Werkzeugpanel



Das Werkzeugpanel im Bearbeitungsmodus ((zur Anschaulichkeit nebeneinander)

Das Werkzeugpanel wird mit T geöffnet und geschlossen. Es werden hier oft benutzte Funktionen gespeichert die auch über Tastenkombinationen verfügbar sind und zwar sind sie entweder im "Spezial"menü das mit W aufgerufen wird oder im Kanten(CtrlE), oder Flächenmenü(CtrlF).

Da diese Schaltfläche dynamisch mit python erstellt wird, kann sie nach belieben umorganisiert oder angepasst werden, sodass gewünschte Funktionen hier verfügbar gemacht werden oder entfernt werden. Siehe dazu [Blender erweitern](#).

Für weitere Informationen über Panele können Sie sich die Sektion über [Referenzpanele](#) ansehen.

Eigenschaftenspanel



Das Eigenschaftenspanel im Bearbeitungsmodus (zur Anschaulichkeit nebeneinander)

Die Eigenschaften Ablage wird mit N geöffnet und geschlossen.

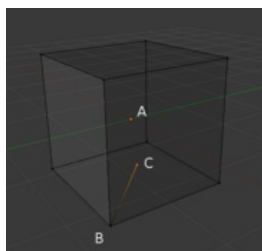
Die Panele in der Eigenschaftenspanel stellen Einstellungen zur Darstellung und zur Bearbeitung des Meshes im Bearbeitungsmodus zur Verfügung die das Mesh nicht direkt verändern, sondern die Anzeigenmodi von meshteilen der 3D Ansicht und die Position der Auswahl bestimmen.

Für weitere Informationen über Panele können Sie sich die Sektion über [Referenzpanele](#) ansehen.

Vertices, Kanten und Flächen

Meshes, geometrischen Figuren, bestehen aus drei Dingen: "Vertices" (quasi Punkte im 3D Raum), "Kanten" und "Flächen". Aus diesen Grundbausteinen kann so ziemlich jedes Objekt konstruiert werden.

Vertices



Vertex Beispiel

Ein Vertex (Einzahl von Vertices) ist ein einziger Punkt im 3D Raum. Mathematische Punkte haben keine Ausdehnung, sie markieren nur einen Ort, daher werden sie auch nicht gerendert und nicht zu verwechseln mit dem Objektzentrum, zu dem wir später kommen, welches größer ist. Das Beispiel zeigt das Objektzentrum A und die Vertices B und C.

Kanten

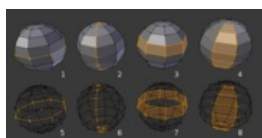
Eine Kante verbindet 2 Vertices mit einer geraden Linie. Diese Kanten können als Drahtgittermodell dargestellt werden. Auch sie werden nicht gerendert.

Flächen

Schlussendlich eine Fläche besteht aus der Fläche die durch die Kanten definiert wird und wird gerendert. Hier ist zu beachten das Dreiecke praktisch sind, da ihre Oberfläche leicht zu berechnen ist. Vierecke hingegen lassen sich zwar schwerer berechnen aber einfacher unterteilen und vorhersagbar verformen. Bei Vierecken sollte darauf geachtet werden, dass alle vier Punkte ungefähr in einer Ebene liegen, sonst kann es zu unschönen Ergebnissen kommen.

Natürlich könnten Objekte auch nur aus Dreiecken zusammengesetzt werden, aber diese können sehr verwirrend sein.

Schleifen



Kanten- und Flächenschleifen

Kanten- und Flächenschleifen bestehen aus aneinander gereihten Flächen und Kanten die aneinander verknüpft sind und irgendwann an ihren Anfang zurückführen wenn man sie verfolgt. Die oberste Reihe im Bild zeigt eine solide und die Reihe ganz unten eine Drahtmodell Ansicht der selben Schleife.

Es ist wichtig zu beachten das die Schleifen 2 und 4 nicht um das Modell herum führen. Schleifen enden an sogenannten Polen, da es keine eindeutige Fortsetzung gibt. Pole sind Vertices die mit drei, fünf oder mehr anderen Vertices verbunden sind. Dementsprechend sind Vertices mit einem, zwei oder vier verbundenen Vertices keine Pole.

Die Schleifen 1 und 3 zeigen das Schleifen zyklisch sind, beim selben Vertex anfangen und aufhören. Schleifen sind ein sehr nützliches Instrument um mit gleichmäßigen Regionen eines Modells in einem bestimmten Bereich zu arbeiten. Mehr Informationen zum arbeiten mit Schleifen befindet sich [hier](#).

Kantenschleifen

Die Schleifen 1 und 3 sind Kantenschleifen. Sie verbinden Vertices so das jedes Vertex genau zwei Nachbarvertices hat.

Kantenschleifen sind sehr wichtig beim erzeugen von organischen Strukturen. Wenn sie richtig verwendet werden erlauben sie Modelle mit verhältnismäßig wenigen Vertices die sehr natürlich aussehen können und verhalten wenn sie animiert werden.

Im Beispiel "Kantenschleifen beim organischen Modellieren kann man sehen das die Kantenschleifen sehr natürlich den Konturen der Haut und der Muskel folgen. Sie sind dichter in Gebieten die sich verformen wenn das Modell animiert wird.

Flächenschleifen

Die logische Fortsetzung von Kantenschleifen sind Flächenschleifen. Schleifen 3 und 4 sind Flächenschleifen und genau wie Kantenschleifen verbinden sie ihre Teile so das alle genau 2 Nachbarn haben und am selben Teil anfangen und aufhören, nur handelt

es sich um Flächen und nicht um Punkte die verbunden sind.

Weitere Informationen wie man Kantenschleifen auswählt befindet sich [hier](#).

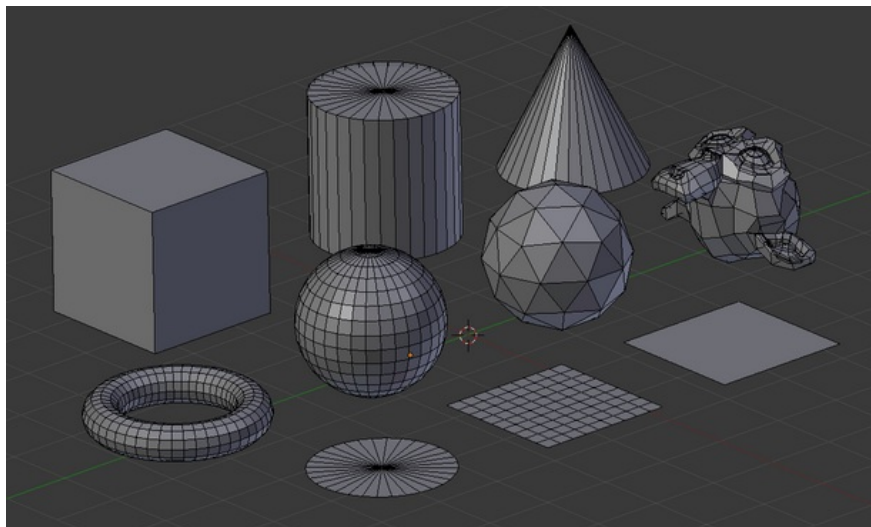
Mesh Grundkörper

Mode: Objektmodus

Hotkey: ⇧ ShiftA

Menu: Hinzufügen » Mesh

Blender hat eine Reihe von Meshes die als Grundlage zum Modelieren oder als Beispiel verwendet werden können.



Blenders zehn standard Meshes

Technisch gesehen ist eine Mesherzeugung eine Meshbearbeitung, daher öffnet sich ein [Kontextmenü](#) in der Werkzeugleiste über das eingestellt werden kann wie der Körper erzeugt werden soll, aber das ist nicht Thema dieser Seite.

Ebene

Eine standard Ebene ist ein Quadrat. Die Oberflächennormale zeigt standardmäßig nach oben.

Würfel

Ein Würfel hat acht Vertices, zwölf Kanten und sechs Flächen.

Kreis

Ein Kreis besteht in Blender aus einer Anzahl von Vertices da Blender mit absoluten Punkten rechnet.

UV Kugel

Da Blender mit absoluten Punkten arbeitet stellt sich die Frage wie eine Kugel mit Punkten beschrieben wird. Die UV Kugel ist eine Möglichkeit das zu tun. Es wird eine Kugel aus vertikal mit einander verbundenen Kreisingen erzeugt, ähnlich wie dem Kugelkoordinatensystem (wie es für Weltkarten um den Globus gelegt wird).

Ikokugel

Eine Ikokugel ist die zweite Möglichkeit mit absoluten Punkten eine Kugel zu beschreiben. Es ist eine polyhedrale Kugel die aus Dreiecken zusammengesetzt ist. Die Anzahl an Unterteilungen bestimmt die Kugelförmigkeit. Es sorgt für eine gleichmäßigere Vertex Dichte auf der Kugel und verhindert so Verzerrungen an den Polen wie sie bei UV Kugeln auftreten.

Warnung!

Die Unterteilung einer Ikokugel erhöht die Rechenleistung exponentiell! Eine Ikokugel mit 10 Unterteilungen hat 5,242,880 Flächen und es ist möglich 500 einzustellen. Es ist gut möglich das Computer (oder zumindest das Programm) bei solchen Datenmengen abstürzen!

Zylinder

Ein Zylinder kann auch als Kreisprisma bezeichnet werden. Dementsprechend besteht er aus zwei miteinander verbundenen Kreisen.

Kegel

Ein Kegel ist eine Pyramide mit einem Kreis als Grundfläche

Torus

Ein Torus ist ein Ring mit einem äußeren und einem inneren Radius die wiederum den Radius der Dicke bestimmen.

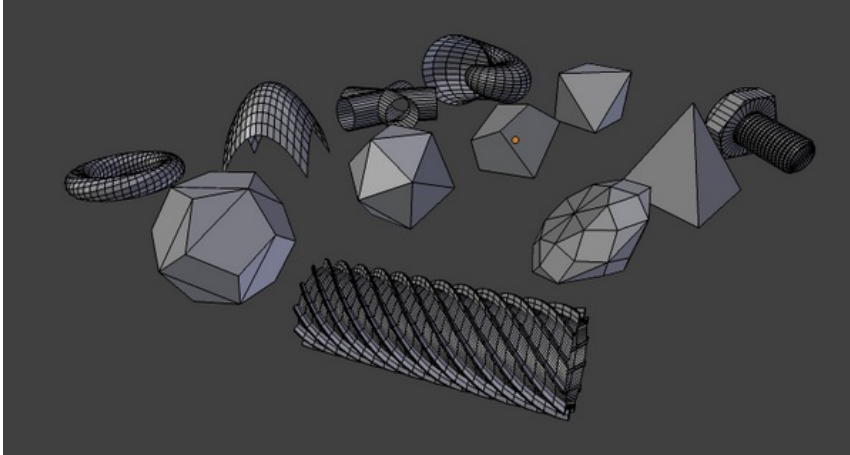
Gitter

Ein Gitter besteht aus durch Kanten verbundenen Vertices.

Affe

Dies ist ein Geschenk von NaN an die Blender community und ist eher ein Easter Egg und Witz, wird aber häufig benutzt um ein Beispiel Objekt, wie auch die [Utah Teetasse](#) oder den [Stanford Hasen](#), zu haben das sich nicht so erwartungsgemäß wie die oben genannten geometrischen Körper verhalten. Der Kopf heißt Suzanne und ist quasi das Blender Maskotchen.

Erweiterungen



Ein paar der durch Erweiterungen verfügbaren Körper.

Es ist natürlich nützlich bestimmte geometrische Grundkörper zu haben die nicht Blenderstandard und sehr schwer von Hand zu modellieren sind. Dafür gibt es mit Skripten erzeugte Modelle die als vorinstallierte Erweiterungen verfügbar sind, aber angeschaltet werden müssen. Einige davon sind:

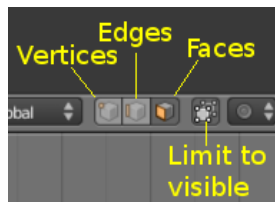
- Mathematisch generierte Oberflächen und Körper
- Edelsteine
- [Landschaft](#)
Fügt eine Landschaft hinzu die als Grundlage dienen kann.
- [Rohrverbindungen](#)
Fügt eine von fünf verschiedenen Rohrverbindungen hinzu.
- [Zahnräder](#)
Adds a gear or a [worm](#) with many parameters to control the shape in the Tool Shelf.

Selektieren von Mesh-Komponenten

Es gibt viele Möglichkeiten Elemente zu selektieren. Welche Selektionswerkzeuge verfügbar sind, hängt davon ab, in welchem Mesh-Selektionsmodus Sie sich befinden. Als Erstes werden wir diese Modi durchgehen und danach die grundlegenden Selektionswerkzeuge betrachten.

Selektionsmodi

Selektionsmode-Kopfwidgets



Edit mode selection buttons

Im Edit mode gibt es drei verschiedene Selektionsmodi. Sie kommen in die verschiedenen Modi durch Auswahl einer der drei Knöpfe in der Werkzeugleiste.

Mit Hilfe dieser Knöpfe können Sie auch in den **Gemischten Mode** wechseln durch ⇧ Shift LMB  beim Anklicken der Knöpfe.

Vertices

Ausgewählte Vertices werden in orange gezeichnet, unselektierte in schwarz und die aktiven oder zuletzt ausgewählte Vertices in weiß.

Edges

In diesem Modus werden die Vertices nicht dargestellt. Stattdessen werden ausgewählte Edges in orange gezeichnet, nicht ausgewählte in schwarz und aktive oder zuletzt ausgewählte in weiß.

Faces

In diesem Modus werden die Faces mit einem Selektionspunkt in der Mitte dargestellt, der zur Auswahl der Faces verwendet wird. Ausgewählte Faces und ihre Selektionspunkte werden in orange gezeichnet, nicht ausgewählte in schwarz und der aktive oder zuletzt ausgewählten Face in weiß.

Fast alle Änderungswerkzeuge sind in allen drei Auswahlmodi vorhanden. So können Rotate, Scale, Extrude usw. in allen Modi benutzt werden. Natürlich tun Rotieren und Skalieren eines *einzelnen* Vertex nichts sinnvolles und einige Werkzeuge sind mehr oder weniger gut verwendbar in einigen Modi.

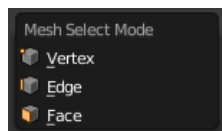
Anmerkung

Die "Modusauswahl"-Knöpfe sind nur im Edit mode sichtbar.

Selektionsmodus Pop-up

Mode: Edit mode

Hotkey: Strg⇐ Tab



Mesh Select Mode menu

Sie können auch einen Auswahl-Modus mit dem Popup-Menü auswählen.

Selektionsmodus » Vertices

Drücke Strg⇐ Tab und wähle Vertices vom Popup-Menü oder drücke Strg⇐ Tab1.

Selektionsmodus » Edges

Drücke Strg⇐ Tab and wähle Edges vom Popup-Menü oder drücke Strg⇐ Tab2.

Selektionsmodus » Faces

Drücke Strg⇐ Tab and wähle Faces vom Popup-Menü oder drücke Strg⇐ Tab3.

Umschalten des Selektionsmodus

Beim Umschalten des Selektionsmodus von Vertices auf Edges und von Edges auf Faces bleiben die selektierten Teile ausgewählt,

wenn sie eine zulässige Menge im neuen Selektionsmodus bilden. Wenn zum Beispiel alle vier Edges eines Faces ausgewählt sind, dann bleibt der Face beim Umschalten vom Edges-Modus in den Faces-Modus ausgewählt. Die ausgewählten Teile, die im neuen Modus keine vollständige Menge bilden, werden abgewählt.

Ausgewählte Teile nach Umschalten des Selektionsmodus

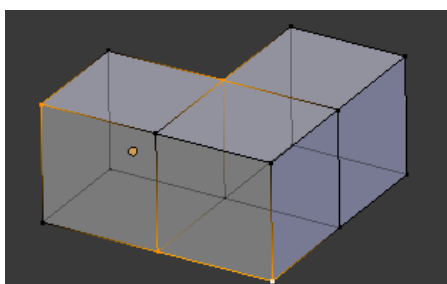
Beim Umschalten des Modus in aufsteigender Reihenfolge (also vom Einfachen zum Komplexeren), von Vertices zu Edges und von Edges zu Faces, bleiben die ausgewählten Teile immer noch ausgewählt, wenn sie ein vollständiges Element im neuen Modus bilden

Wenn zum Beispiel alle vier Edges eines Faces ausgewählt sind, dann bleibt das Face ausgewählt, wenn vom Edges-Modus zum Faces-Modus umgeschaltet wird. Alle ausgewählten Teile, die im neuen Modus keine vollständige Menge bilden, werden abgewählt.

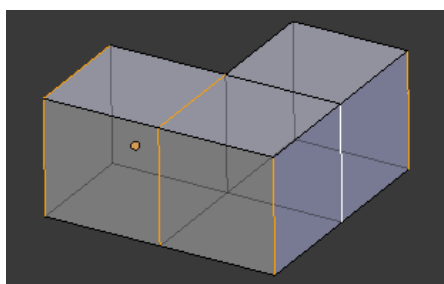
Jedoch beim Umschalten in "absteigender" Reihenfolge (also vom Komplexen zum Einfachen) werden alle Elemente, die das "höhere" Element (wie ein Face) bilden, ausgewählt(die vier Vertices oder Edges eines Vierecks zum Beispiel).

Beim Festhalten von Strg beim Auswählen eines höheren Auswahlmodus, werden alle Elemente hinzugefügt, die die augenblickliche Auswahl berühren, auch wenn die Auswahl kein vollständiges höheres Element bildet.

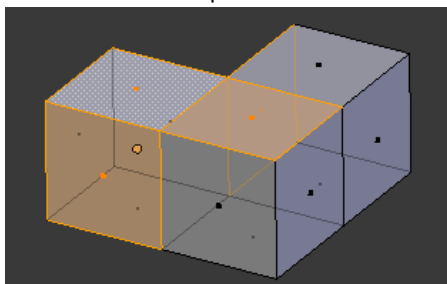
Siehe (*Vertices-Modus-Beispiel*), (*Edges-Modus-Beispiel*), (*Faces-Modus-Beispiel*) and (*Mixed-Modus-Beispiel*) für Beispiele der verschiedenen Modi.



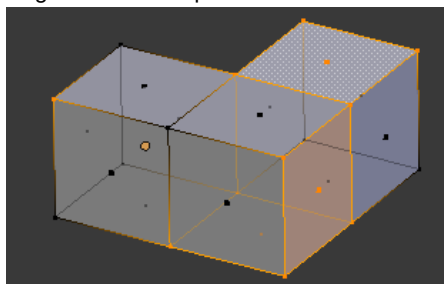
Vertices mode example.



Edges mode example.



Faces mode example.



Mixed mode example.

Basic Selection

Mode: Edit mode


Hotkey: RMB  and ⇧ Shift RMB 

Die meistgenutzte Methode zur Auswahl eines Elements ist RMB  darauf anzuwenden, dies ersetzt die derzeitige Auswahl durch das neue Element.

Hinzufügen zur Auswahl

Zum Hinzufügen zur bestehenden Auswahl halten Sie ⇧ Shift während des Rechtsklicks gedrückt. Wiederholtes Klicken auf ein ausgewähltes Element wählt es ab.

Wie im Object-Modus, gibt es ein eindeutiges *aktives* Element, das in größerer Helligkeit dargestellt (in der Regel das zuletzt ausgewählte Element). Abhängig vom benutzten Werkzeug kann dieses Element sehr wichtig sein!

Beachten Sie, dass es keine Möglichkeit gibt zwischen sich überdeckenden Elementen zu wählen (wie der Alt RMB -Klick im Object-Modus). Sind Sie jedoch im soliden, schattierten- oder texturierten Viewport-Schattierungs-Modus (nicht in Bounding-Box oder Wireframe), dann gibt es rechts der Selektionsmodi einen vierten Knopf, der wie ein Würfel aussieht.

Ist dieser aktiviert, so wird die Auswahl auf sichtbare Elemente eingeschränkt (als ob das Objekt fest wäre) und bewahrt sie vor dem unabsichtlichen Auswählen, Verschieben, Löschen oder anderweitigem Ändern von rückseitigen oder versteckten Elementen.

Auswählen von Elementen eines Bereichs

Mode: Edit-Modus

Hotkey: B, C, und Strg LMB  Klicken und Ziehen

Die Bereichsauswahl ermöglicht es Gruppen von Elementen innerhalb einer 2D-Region ihres 3D-Views auszuwählen. Der Bereich kann ein Kreis oder ein Rechteck sein. Der Kreisbereich ist nur im Edit mode verfügbar. Der Rechteckbereich, auch

“*Rahmenauswahl*”, ist im Edit-Modus und im Object-Modus verfügbar.


Anmerkung

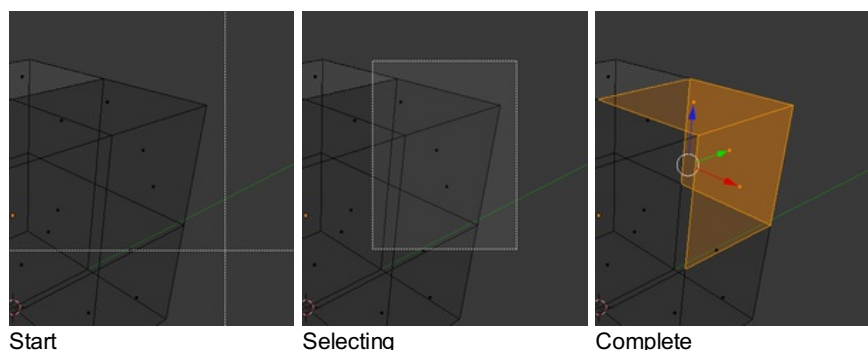
Was mit diesen beiden Werkzeugen ausgewählt wird, wird von der Limit Selection to visible-Eigenschaft (verfügbar im 3D viewport) in Solid Viewport Shading Mode beeinflusst.



Zum Beispiel:

1. im Festkörper-Modus und im Face-Auswahlmodus werden alle faces *innerhalb* des Auswahlbereichs selektiert
2. im Wireframe-Modus und Face-Auswahlmodus werden nur Faces ausgewählt, deren Selektionspunkt sich innerhalb des Auswahlbereichs befindet.


Rechteckbereich (Rahmenauswahl)

Die Rahmenauswahl ist sowohl im Edit mode als auch im Object mode verfügbar. Zum Aktivieren des Werkzeugs benutzen Sie B. Benutzen sie Border Select um eine Gruppe von Objekten durch das Zeichnen eines Rechtecks auszuwählen, während sie LMB  drücken. Dabei wählen Sie alle Objekte aus, die innerhalb dieses Rechtecks liegen oder von dem Rechteck angeschnitten werden. Liegt irgendein Objekt in der Gruppe, das zuletzt aktiv war, so wird es ausgewählt *und* aktiv.




Bei (*Start*), wurde die Rahmenauswahl eingeleitet und wird durch einen gepunkteten Kreuzzeiger gekennzeichnet. Bei (*Selecting*) wird der *Selektionsbereich* ausgewählt durch Zeichnen eines Rechtecks mit gedrücktem LMB . Der Selektionsbereich überdeckt nur die Selektionspunkte von 3 Faces. Schließlich ist die Auswahl abgeschlossen bei Loslassen von LMB ; siehe (*Complete*).

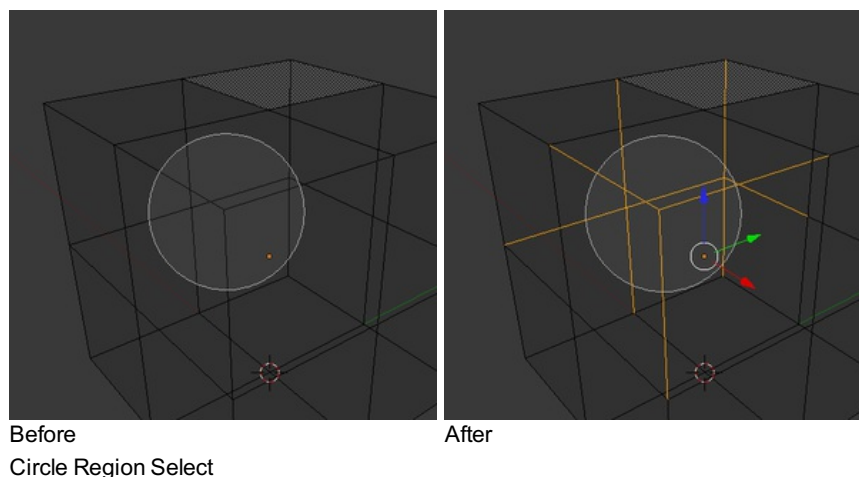
Anmerkung


Die Bereichsauswahl fügt eine Auswahl zu der bereits bestehenden Auswahl hinzu. Um also nur den Inhalt des Rechtecks auszuwählen, muss man vorher alles mit A abwählen. Außerdem kann man mit MMB  alle Objekte innerhalb des Rechtecks abwählen.

Circular region

This selection tool is only available in Edit mode and can be activated with C. Once in this mode the cursor changes to a dashed cross-hair with a 2D circle surrounding it. The tool will operate on whatever the current select mode is. Clicking or dragging with the LMB , when elements are inside the circle, will cause those elements to be selected.

You can enlarge or shrink the circle region using + NumPad and - NumPad, or the Wheel .



(*Circle Region Select*) is an example of selecting edges while in Edge Select Mode. As soon as an edge intersects the circle the edge becomes selected. The tool is interactive such that edges are selected while the circle region is being dragged with the LMB .


If you want to de-select elements either hold MMB  or Alt LMB  and begin clicking or dragging again.

For Faces select mode, the circle must intersect the face indicators usually represented by small pixel squares; one at the center of each face.

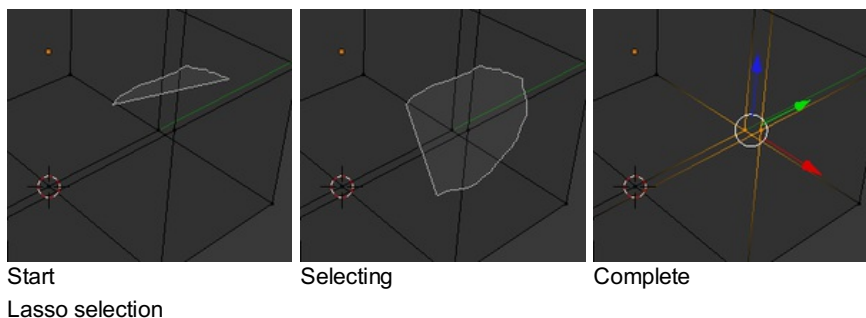
To exit from this tool click RMB , or hit the Esc key.

Lasso region

Lasso select is similar to Border select in that you select objects based on a region, except Lasso is a hand-drawn region that generally forms a circular/rounded shaped form; kind of like a lasso.

Lasso is available in either Edit Mode or Object Mode. To activate the tool use the Ctrl LMB  while dragging. The one difference between Lasso and Border select is that in Object mode, Lasso only selects objects where the lasso region intersects the objects centre.

To de-select use Ctrl⇧ Shift LMB  while dragging.



(*Lasso selection*) is an example of using the Lasso select tool in Vertex Select Mode.

Additional Selection Tools

The select menu in edit mode contains additional tool for selecting components:

Basic Tools

Select All/Select None A

Select all or none of the mesh components.

Invert Selection CtrlI

Selects all components that are not selected, and deselect currently selected components.

More CtrlNum+

Propagates selection by adding components that are adjacent to selected elements.

Less CtrlNum-

Deselects components that form the bounds of the current selection

Advanced Tools

Mirror

Select mesh items at the mirrored location.

Linked CtrlL

Selects all components that are connected to the current selection.

Select Random

Selects a random group of vertices, edges, or faces, based on a percentage value.

Select Every N Number of Vertices

Selects vertices that are multiples of N.

Select Sharp Edges

This option will select all edges that are between two faces forming an angle less than a given value, which is asked you *via* a small pop-up dialog. The lower is this angle limit, the sharper will be the selected edges. At **180°**, **all** "manifold" (see below) edges will be selected.

Linked Flat Faces (Ctrl⇧ ShiftAltF)

Select connected faces based on a threshold of the angle between them. This is useful for selecting faces that are planar.

Select Non Manifold (Ctrl⇧ ShiftAltM)

Selects vertices that are not completely bound by geometry, including border edges, floating edges, and orphan vertices. Only available in Vertex mode.

Interior

Select faces where all edges have more than 2 faces.

Side of Active

Selects all data on the mesh in a single axis

Tris

Select all triangles in the mesh

Quads

Select all quads in the mesh

Loose

Select all vertices or edges that do not form part of a face.

Select Similar

Mode: Edit mode

Hotkey: ⇧ ShiftG

Menu: Select » Similar to Selection...

Select components that have similar attributes to the ones selected, based on a threshold that can be set in tool properties after activating the tool. Tool options change depending on the selection mode

Vertex Selection Mode

Normal

Selects all vertices that have normals similar to those currently selected.

Amount of Vertices in Face

Selects all vertices that belong to exactly the same number of faces as those already selected. For example, assuming you have only one selected vertice, if this one does not belong to any face, this option will select all vertices not involved in faces.

Vertex Groups

Selects all vertices sharing one or more [vertex groups](#) with those already selected.

Edge Selection Mode

Length

Selects all edges that have a similar length as those already selected.

Direction

Selects all edges that have a similar direction (angle) as those already selected.

Amount of Vertices in Face

Selects all edges that belong to exactly the same number of faces as those already selected. This is the same as for Vertex select mode described above.

Face Angles

Selects all edges that are between two faces forming a similar angle, as with those already selected.

Crease

Selects all edges that have a similar Crease value as those already selected. The Crease value is a setting used by the [Subsurf Modifier](#).

Seam

Selects all edges that have the same Seam state as those already selected. Seam is a two-states setting used in [UV-texturing](#).

Sharpness

Selects all edges that have the same Sharp state as those already selected. Sharp is a two-state setting (a flag) used by the [EdgeSplit Modifier](#).

Face Selection Mode

Material

Selects all faces that use the same material as those already selected.

Image

Selects all faces that use the same UV-texture as those already selected (see [UV-texturing](#) pages).

Area

Selects all faces that have a similar area as those already selected.

Perimeter

Selects all faces that have a similar perimeter as those already selected.

Normal

Selects all faces that have a similar normal as those selected. This is a way to select faces that have the same orientation (angle).

Co-planar

Selects all faces that are (nearly) in the same plane as those selected.

Selecting Loops

You can easily select loops of components:

Edge Loop and Vertex Loop Selection

Mode: Edit mode → Vertex or Edge select mode

Hotkey: Alt RMB  or CtrlE → Select » Edge Loop

Menu: Select » Edge Loop or Mesh » Edges » Edge Loop

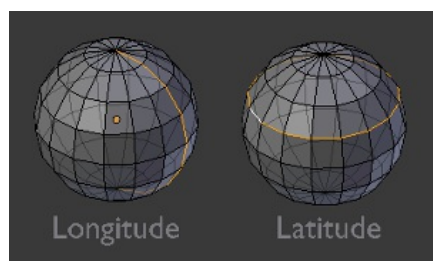
Holding Alt while selecting an edge selects a loop of edges that are connected in a line end to end, passing through the edge under the mouse pointer. Holding Alt+Shift while clicking adds to the current selection.

Edge loops can also be selected based on an existing edge selection, using either Select » Edge Loop, or the Edge Loop Select option of the Edge Specials menu (CtrlE).

Vertex mode

In Vertex select mode, you can also select edge loops, by using the same hotkeys, *and clicking on the edges* (not on the vertices).

Example



Longitudinal and latitudinal edge loops.

The left sphere shows an edge that was selected longitudinally. Notice how the loop is open. This is because the algorithm hit the vertices at the poles and terminated because the edge at the pole connects to more than three other edges. However, the right sphere shows an edge that was selected latitudinally and has formed a closed loop. This is because the algorithm hit the first edge that it started with.

Face Loop Selection

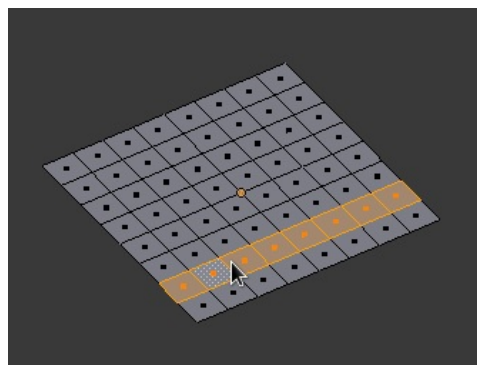
Mode: Edit mode → Face or Vertex select modes

Hotkey: Alt RMB 


In face select mode, holding Alt while selecting an **edge** selects a loop of faces that are connected in a line end to end, along their opposite edges.

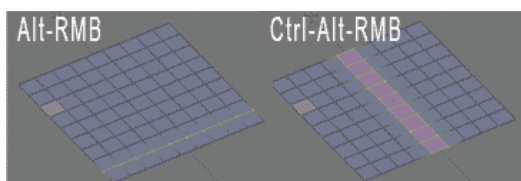
In vertex select mode, the same can be accomplished by using CtrlAlt to select an edge, which selects the face loop implicitly.

Examples



Face loop selection.

This face loop was selected by clicking with Alt RMB  on an edge, in face select mode. The loop extends perpendicular from the edge that was selected.



Alt versus CtrlAlt in vertex select mode.

A face loop can also be selected in Vertex select mode, see (*Alt versus CtrlAlt in vertex select mode*). The edges selected in the grid labeled “Alt-RMB” is a result of selecting and edge loop versus selecting an edge ring. Because in vertex select mode, selecting opposite edges of a face implicitly selects the entire face, the face loop has been selected implicitly.

Note that in these cases the generated result of the algorithm was *vertices* because we were in Vertex select mode. However, had we had been in Edge select mode, the generated result would have been selected edges.

Edge Ring Selection

Mode: Edit mode → Edge select mode

Hotkey: CtrlAlt RMB  or CtrlE → Select » Edge Ring

Menu: Select » Edge Ring or Mesh » Edges » Edge Ring

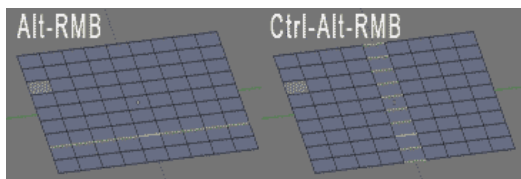
In Edge select mode, holding CtrlAlt while selecting an edge selects a sequence of edges that are not connected, but on opposite sides to each other continuing along a [face loop](#).

As with edge loops, you can also select edge rings based on current selection, using either Select » Edge Ring, or the Edge Ring Select option of the Edge Specials menu (CtrlE).

Vertex mode

In Vertex select mode, you can use the same hotkeys when *clicking on the edges* (not on the vertices), but this will directly select the corresponding face loop...

Example



A selected edge loop, and a selected edge ring.

In (*A selected edge loop, and a selected edge ring*), the same edge was clicked on but two different “groups of edges” were selected, based on the different commands. One is based on edges during computation and the other is based on faces.

Loop to Region and Region to Loop

Mode: Edit mode → Edge select mode

Hotkey: CtrlE » 9 NumPad and CtrlE → Select » Loop to Region/Region to Loop

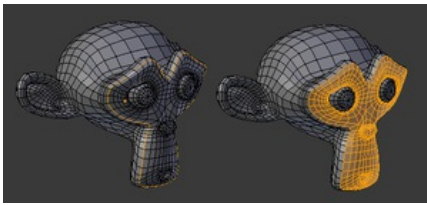
Menu: Select » Loop to Region/Region to Loop or Mesh » Edges » Loop to Region/Region to Loop

Loop to Region examines the current set of selected edges and separates them into groups of “loops” that each bisect the mesh into two parts. Then for each loop it selects the smaller “half” of the mesh. Even though it works in Vertex and Face select mode, the results using these might be strange – you should stick to Edge select mode...

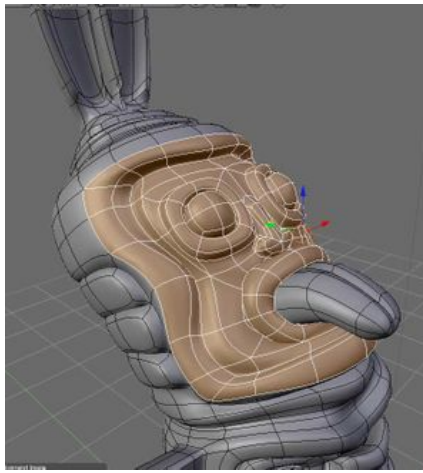
Region to Loop is the “logical inverse” of Loop to Region, based on all regions currently selected, it selects only the edges at the border of these regions. It can operate in any select mode, but always switch in Edge select mode when applied.

All this is much more simple to illustrates with examples:

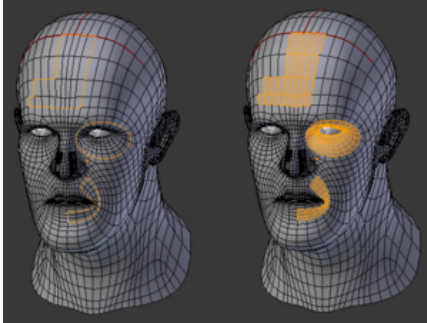
Example: Loop to Region



Selection.



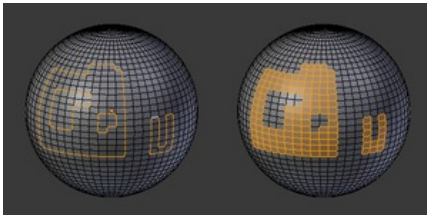
Loop to Region.



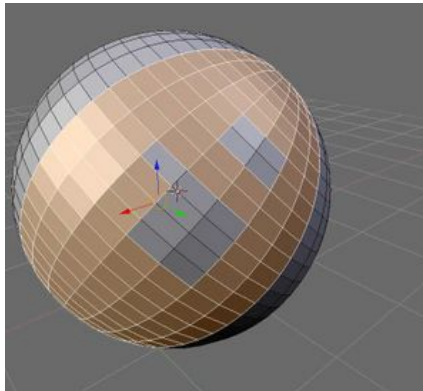
Selection.



This tool handles multiple loops fine, as you can see.

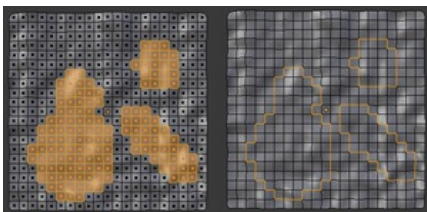


Selection.

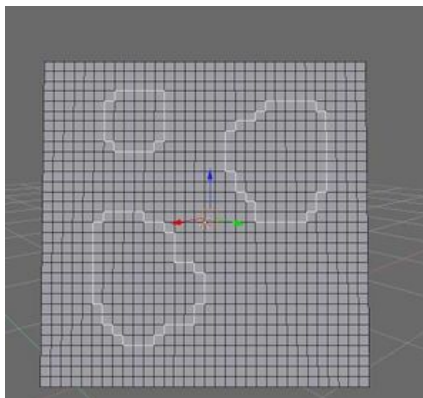


This tool handles "holes" just fine as well.

Example: Region to Loop



Selection.



This is the "logical inverse" of loop to region.

Auswählbare Elemente

Wie auf der [Mesh Struktur Seite](#) einsehbar ist, bestehen Meshes aus verschiedenen Grundkörpern, die allerdings nur unterschiedlich dargestellte identische Datenpunkte sind.

Da man ein Mesh in diesen unterschiedlichen Arten darstellen kann, kann man es auch so auswählen. So werden wenn man eine Fläche auswählt auch die dazugehörigen Kanten und Vertices ausgewählt. Das ist sehr wichtig wenn man versucht eine Funktion auf Elemente anzuwenden die diese gar nicht unterstützen, also eine Flächenverändernde Funktion auf Vertices oder Kanten.

Es kann nur ein Typ von Element zur gleichen Zeit ausgewählt werden, abhängig vom aktuellen Auswahlmodus. Ausgewählte Elemente werden aber beibehalten wenn der neue Auswahlmodus bei einem Wechsel das zulässt.

Ausgewählte Elemente werden in orange, nicht ausgewählte schwarz und das aktive Element in weiß dargestellt. Diese Farben können in den [Benutzereinstellungen](#) geändert werden.

Auswahlmodi

Es gibt zwei Wege zwischen den Auswahlmodi zu wechseln:

- Strg⇌ Tab
- Die Knöpfe in der Kopfzeile der 3D Ansicht.



Bearbeitungsmodus
Auswahlknöpfe

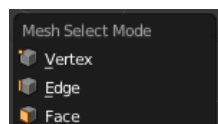
Auswahl Elemente können mit ⇧ Shift kombiniert werden.

Moduspopup

Mode: Bearbeitungsmodus

Hotkey: Ctrl⇌ Tab

Im Bearbeitungsmodus gibt es drei verschiedene Auswahlmodi für Meshes (siehe Bild).



Auswahlmodimenü

Select Mode » Vertices

Es sind Vertices auswählbar und es können bei Kombination von entsprechenden Vertices auch Kanten und Flächen ausgewählt werden.

Select Mode » Kanten

In diesem Modus werden Vertices nicht mehr dargestellt aber ausgewählt wenn sie eine ausgewählte Kante bilden. Stattdessen werden nur noch Kanten und Flächen dargestellt. Es können Kanten und bei Kombination von entsprechenden Kanten auch Flächen ausgewählt werden.

Select Mode » Flächen

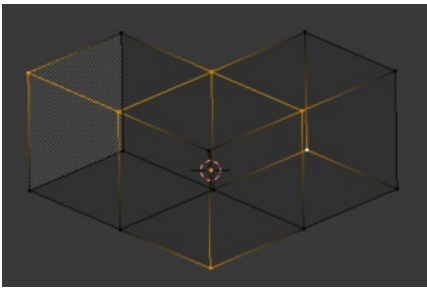
In diesem Modus sind nur noch Flächen einzeln auswählbar, diese Auswahl enthält dann alle zur Fläche gehörenden Kanten und Vertices. Es wird ein Auswahlpunkt in der Mitte der Fläche angezeigt.

Fast alle Werkzeuge sind in den jeweiligen Modi verfügbar, sie machen nur manchmal keinen Sinn, wie Rotation von einzelnen Vertices

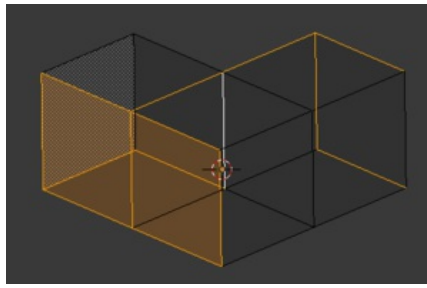
Auswahl bei Moduswechsel

Wenn in einen komplexeren Modus gewechselt wird (Vertex->Kante->Fläche) bleiben alle im neuen modus gültigen Elemente ausgewählt, einzelne Vertices oder Kanten die keine Flächen bilden, werden ggf. abgewählt. Umgekehrt werden aber alle ausgewählten Elemente in Elemente des niedrigeren Typs umgewandelt und bleiben ausgewählt.

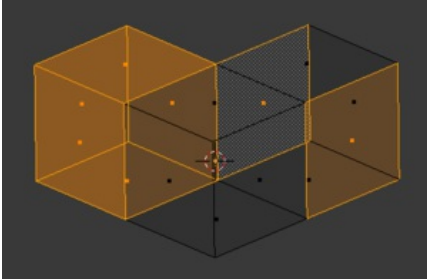
See (*Vertices mode example*), (*Edges mode example*), (*Faces mode example*) and (*Mixed mode example*) for examples of the different modes.



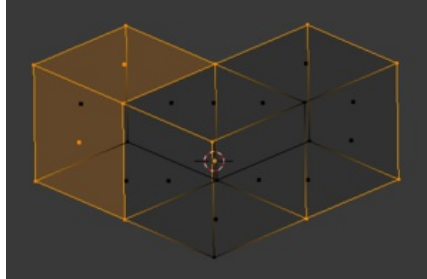
none Vertexmodus



Kantenmodus



Flächenmodus



gemischter Modus


Kanten auswählen

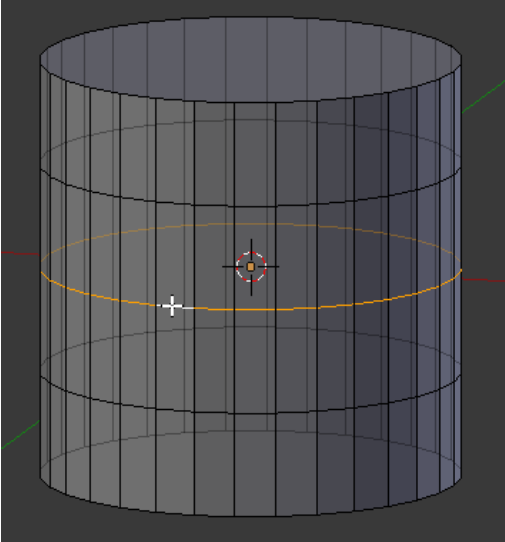
Kantenschleifen auswählen

Mode: Bearbeitungsmodus

Hotkey: Alt RMB  - oder ⇧ ShiftAlt RMB  zum hinzufügen zur Auswahl oder Abwählen

Menu: Auswahl » Kantenschleife

Kantenschleifen können auf zwei Arten ausgewählt werden: durch auswählen einer Kante und benutzen des Auswahlmenüs oder durch drücken von Alt RMB  in der Nähe einer Kante.



Alt auf Linux


Alt wird in manchen Linux distros vom Windowsmanager benutzt. Es kann also zu Problemen kommen.

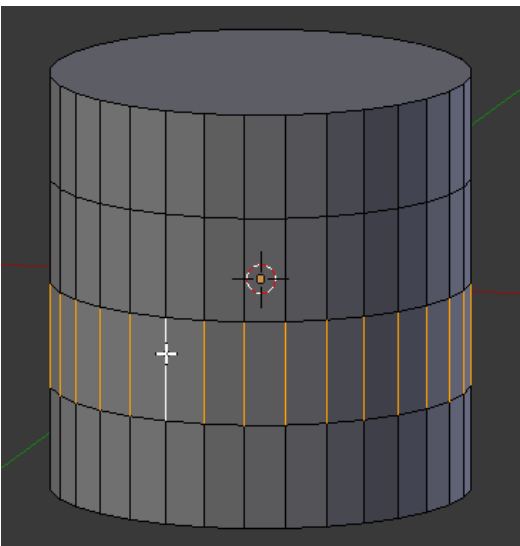
Kantenringe

Mode: Bearbeitungsmodus


Hotkey: AltCtrl RMB  - oder ⇧ ShiftAltCtrl RMB  zum hinzufügen zur Auswahl oder Abwählen

Menu: Auswahl » Kantenring

Kantenringe werden ähnlich ausgewählt: Bei ausgewählter Kante mit der Menüfunktion Auswahl » Kantenring oder durch drücken von AltStrg RMB  in der Nähe einer Kante.




Flächenauswahl

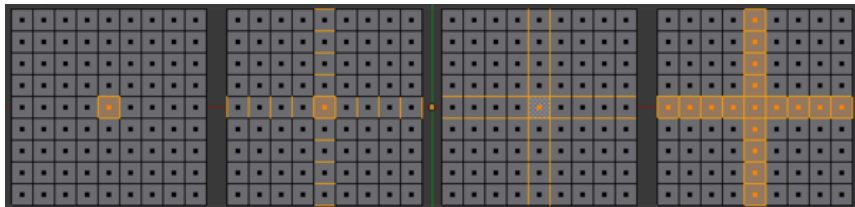
Die Auswahl funktioniert wie üblich mit RMB .

Flächenschleifenauswahl

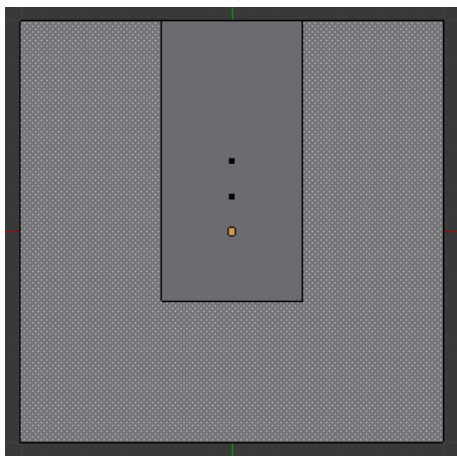
Mode: Bearbeitungsmodus

Hotkey: Alt RMB  - oder ⇧ ShiftAlt RMB  zum hinzufügen zur Auswahl oder zum Abwählen

Flächenschleifen werden genau wie Kantenschleifen ausgewählt, durch drücken von Alt RMB  in der Nähe einer Kante die zwei Flächen der gewünschten Schleife gemein ist.



Ngone im Flächenauswahlmodus



Wie bereits bekannt werden Flächen in der Mitte markiert. Bei Ngonen kann das teilweise Verwirrend sein. Das Ngon in U Form hat seine Markierung neben der Markierung der Fläche die an dieser Stelle existiert. Glücklicherweise muss zur Flächenauswahl die Fläche ausgewählt werden und nicht der Markierungspunkt.

Vertex Werkzeuge

Diese Seite behandelt die vielfältigen Werkzeug die im Mesh » Vertices Menü zu finden sind. Sie funktionieren größtenteils auf Vertexbasis, manche sind aber auch mit Kanten oder Flächen verfügbar

Verschmelzen

Verschmelzen von Vertices

Mode: Bearbeitungsmodus

Hotkey: AltM

Menu: Mesh » Vertices » Verschmelzen, Spezial » Merge or Vertex Spezial » Merge

Dieses Werkzeug verbindet alle ausgewählten Vertices in einem wobei alle Beziehungen erhalten bleiben. An welchem Punkt das neue Vertex entstehen soll kann festgelegt werden:

Am Ersten

Das neue Vertex wird am Ort des zuerst ausgewählten platziert (nur verfügbar im Vertexauswahlmodus).

Am Letzten

Das neue Vertex wird am Ort des zuletzt ausgewählten platziert (nur verfügbar im Vertexauswahlmodus).

Im Zentrum

Das neue Vertex wird im gemeinsamen Zentrum platziert.

Im Cursor

Das neue Vertex wird an der Position des 3D Cursors platziert.

Kollabieren

Verschmilzt jede zusammenhängende Auswahl in ihrem Zentrum.

Automatisches Verschmelzen

Mode: Bearbeitungsmodus

Menu: Mesh » automatisches Verschmelzen

Diese Funktion ermöglicht das automatische verschmelzen von Vertices wenn diese sich näher als ein angegebenes Minimum sind.

Doppelte Entfernen

Mode: Bearbeitungsmodus

Panel: Bearbeitungskontext → Mesh Werkzeuge

Hotkey: W » 4 oder CtrlV » Doppelte Entfernen

Menu: Mesh » Vertices » Doppelte Entfernen, Spezial » Doppelte Entfernen oder Vertex Spezial » Doppelte Entfernen

Doppelte zu entfernen kann sehr nützlich sein um ein Mesh aufzuräumen da sich zu nahe stehende Vertices verschmolzen werden. Dadurch können zum Beispiel Überbleibsel von fehlgeschlagenen Operationen ohne Aufwand entfernt werden. Für einen ähnlichen Effekt kann auch der [Dezimierungsmodifikator](#) benutzt werden.

Verschmelzungsentfernung

Stellt die Entfernung in Blendereinheiten ein ab der Verschmolzen werden soll.

Unausgewählte

Erlaubt das verschmelzen von nicht ausgewählten Vertices mit ausgewählten.

Teilungswerkzeuge

Zerreißen

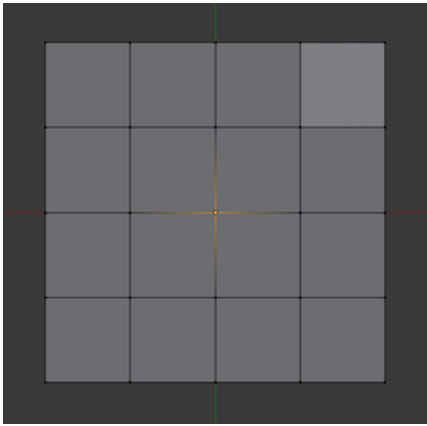
Mode: Bearbeitungsmodus

Hotkey: V

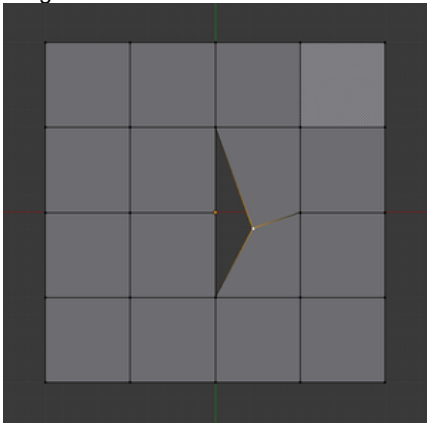
Menu: Mesh » Vertices » Zerreißen

Zerreißen erzeugt ein Loch im Mesh durch kopieren ausgewählter Vertices und Edges und durch das lösen und erzeugen von Verbindungen. Das Ergebnis ist das die Kopie einige Verbindungen des Originals übernimmt und dafür im Original löscht:

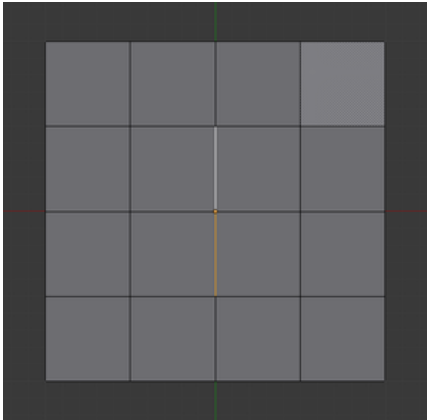
Beispiele



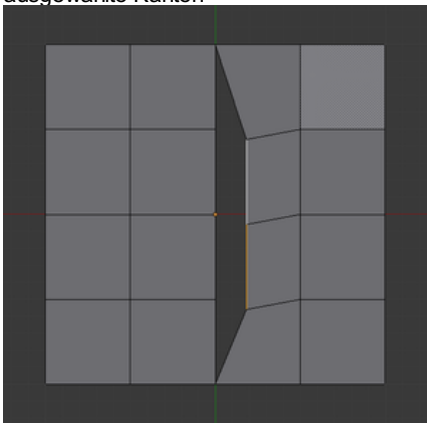
ausgewähltes Vertex



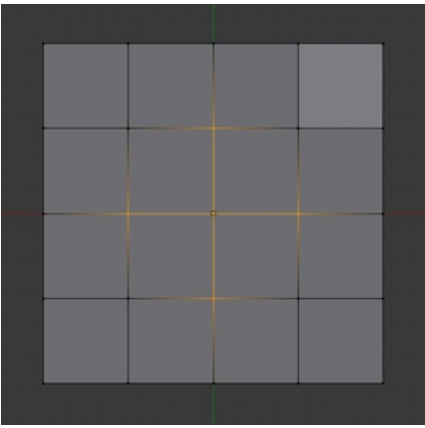
durch zerreißen erzeugtes Loch



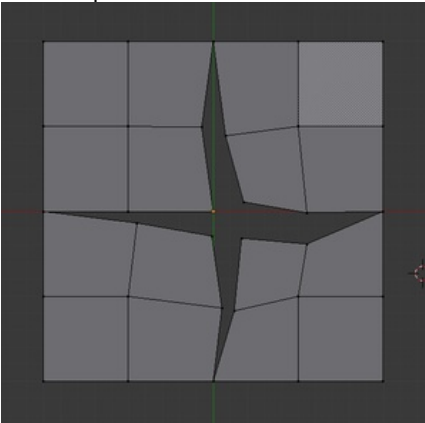
ausgewählte Kanten



durch zerreißen erzeugtes Loch



eine komplexe Auswahl von Vertices



durch zerreißen erzeugtes Loch

Grenzen von Zerreißen

Zerreißen funktioniert nur wenn Kanten oder Vertices ausgewählt sind.

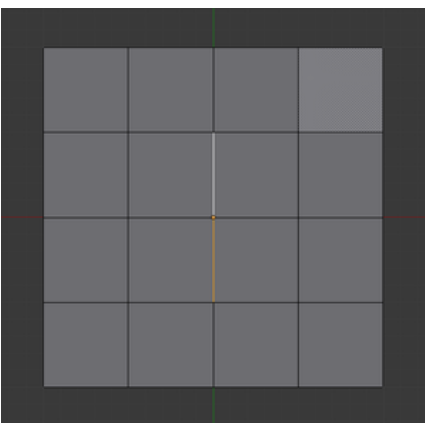
Zerreißen und Füllen

Mode: Bearbeitungsmodus

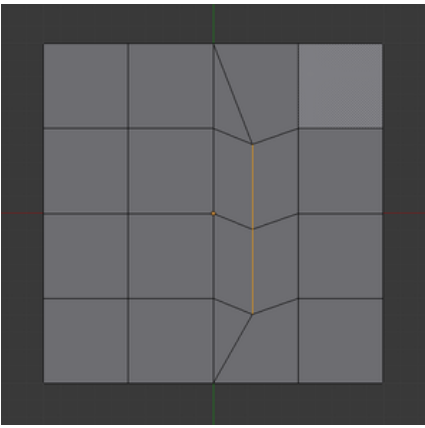
Hotkey: AltV

Menu: Mesh » Vertices » Zerreißen und Füllen

Funktioniert wie Zerreißen, füllt aber das entstehende Loch.



ausgewählte Kanten



Ergebnis von zerreißen und Füllen

Herauslösen

Mode: Bearbeitungsmodus

Hotkey: Y

Menu: Mesh » Vertices » Teilen

Dieses Werkzeug löst ein Vertex, eine Kante oder eine Fläche aus einem Mesh heraus ohne das Mesh dabei zu verändern. Allerdings wird eine Ursprungsfläche dabei entfernt. Es funktioniert also wie ein Duplikat, mit dem Unterschied das originale Flächen aufgelöst werden.

Abteilen

Mode: Bearbeitungsmodus

Hotkey: P

Menu: Mesh » Vertices » Abteilen

Dies verschiebt den ausgewählten Teil in ein neues Objekt. Objekteigenschaften wie zum Beispiel Modifikatoren werden dabei mitkopiert Siehe außerdem hier [hier](#).

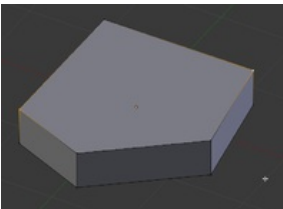
Vertices Verbinden

Mode: Bearbeitungsmodus

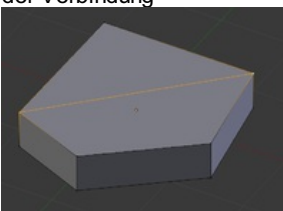
Hotkey: J

Menu: Mesh » Vertices » Vertices verbinden oder StrgV » Vertex Connect

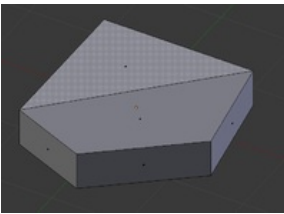
Vertices verbinden verbindet zwei Vertices die Teil einer gemeinsamen Fläche sind, sodass die Fläche geteilt wird.



Ausgewählte Vertices vor der Verbindung



Nach der Verbindung



Zwei Flächen die erzeugt wurden.

Vertex Verschieben

{{RefBox

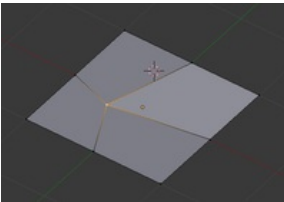
|mode=Bearbeitungsmodus

Vertex verschieben verschiebt ein Vertex entlang einer Kante. Nach aktivieren der Funktion und auswahl der Kante gibt es drei zusätzliche Funktionen zum normalen Verschieben:

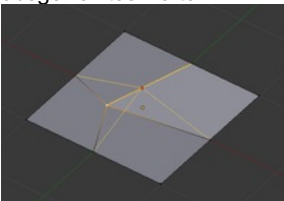
Springe zum Mittelpunkt ⇧ Shift

Springe zum Endpunkt Alt

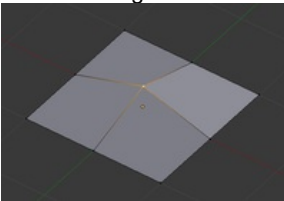
Springe zu und verschmelze mit dem Endpunkt Control



ausgewähltes Vertex



Positionierung des Vertex



Neu positionierter Vertex

Glätten

Mode: Bearbeitungsmodus

Panel: Bearbeitungskontext → Mesh Werkzeuge

Hotkey: CtrlV » Vertex glätten

Menu: Mesh » Vertices » Glätten, Spezial » Glätten or Vertex Spezial » Glätten

Wendet das [Glätten Werkzeug](#) einmal an.

Vertexunterordnung

Mode: Bearbeitungsmodus

Hotkey: CtrlP

Menu: Mesh » Vertices » Vertexunterordnung

Ordnet alle anderen ausgewählten Objekte dem Vertex unter wie [hier](#) beschrieben.

Haken hinzufügen

Mode: Bearbeitungsmodus

Hotkey: CtrlH

Menu: Mesh » Vertices » Haken hinzufügen

Fügt einen [Hakenmodifikator](#) hinzu.

Formübergang, Propagate Shapes

Mode: Bearbeitungsmodus

Hotkey: W » AltFormübergang oder CtrlV » Formübergang, und W » AltShape propagate or CtrlV » Shape Propagate

Menu: (Vertex) Spezial » Formübergang und Vertex Spezial » Shape Propagate

Dies sind Werkzeuge für [Formen \(shape keys\)](#).

Spiegeln


Mode: Bearbeitungsmodus

Hotkey: CtrlM

Menu: Mesh » Spiegeln » gewünschte Achse

Das Spiegel Werkzeug spiegelt eine Auswahl über eine ausgewählte Achse.

Das Spiegelwerkzeug ist dem Spiegel im Objektmodus sehr ähnlich. Es führt die gleiche Operation wie das Skalieren entlang einer entsprechenden Achse über den Pivot Punkt mit dem Faktor -1. Es ist unter Umständen allerdings manchmal nützlicher.

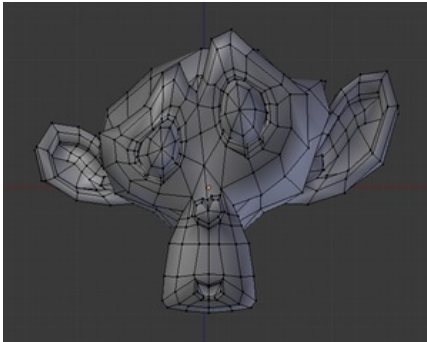
Dieses Werkzeug benutzt [Achsenbeschränkung](#). Es kann aber auch mit der LMB  in die gewünschte Richtung geklickt werden.

Spiegelachse

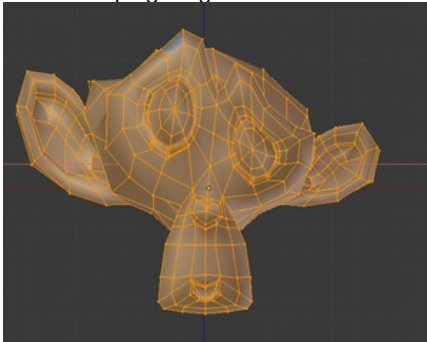
Für jede Transformationsrichtung kann eine Achse ausgewählt werden entlang der gespiegelt werden soll.

Pivotpunkt

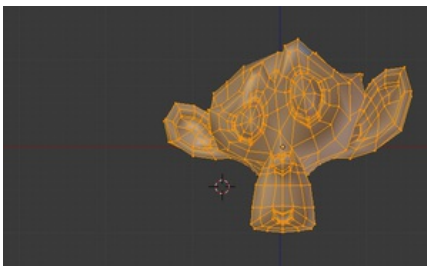
Der [Pivotpunkt](#) legt das Zentrum der Spiegelung fest.



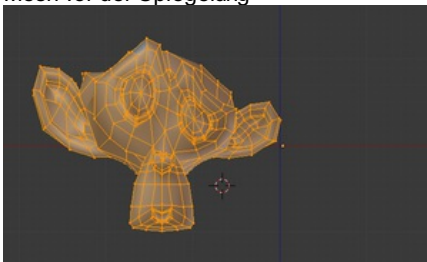
Mesh vor Spiegelung



Mesh nach Spiegelung



Mesh vor der Spiegelung



Mesh nach der Spiegelung mit dem 3D

Cursor als Pivotpunkt

Glätten

Mode: Bearbeitungsmodus

Panel: Mesh Werkzeuge (Bearbeitungskontext, F9)

Hotkey: CtrlV » Vertex Glätten

Menu: Mesh » Vertices » Vertex Glätten

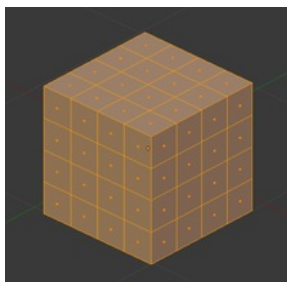
Dieses Werkzeug glättet ausgewählte Elemente durch berechnen des Durchschnitts des Winkels zwischen den Flächen. Das [Funktionsmenü](#) bietet folgende Einstellungen:

Glättungsanzahl

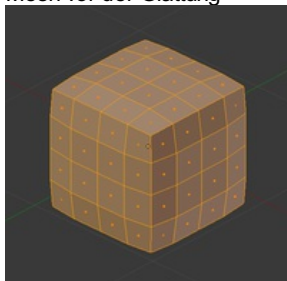
Anzahl an durchgeführten Berechnungsschritten der Glättung.

Achsen

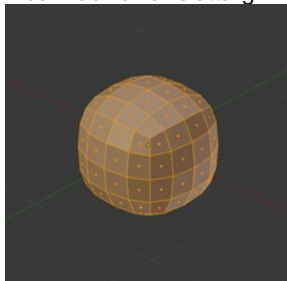
Beschränkt die Glättung auf bestimmte Achsen.



Mesh vor der Glättung



Mesh nach einer Glättung



Mesh nach 10 Glättungen

Laplace Glättung

Mode: Bearbeitungsmodus

Hotkey: W » Laplaceglättung

Siehe den [Laplace Glättungs Modifizierer](#) für mehr Details.

Laplace Glättung benutzt einen alternative Glättungsalgorithmus der die Form des Meshes besser bewahrt. Laplace Glättung gibt es als nicht zerstörenden Meshmodifizierer.

Hinweis

Der [Glättungsmodifizierer](#), der auf Vertexgruppen beschränkt werden kann ist eine alternative zum Glättungswerkzeug.

Echtes Glätten gegenüber glatten Schatten

Dieses Werkzeug ist nicht mit den [Schattenoptionen](#) zu verwechseln, sie haben unterschiedliche Wirkungen! Das Glättungswerkzeug verändert das Mesh ansich während die Schattenoptionen nur einstellen wie es aussieht. Die Schattenoptionen erzeugen nur die Illusion der Glattheit, ohne das Mesh zu verändern.

Extrudieren

Extrudieren von Flächen

Mode: Bearbeitungsmodus

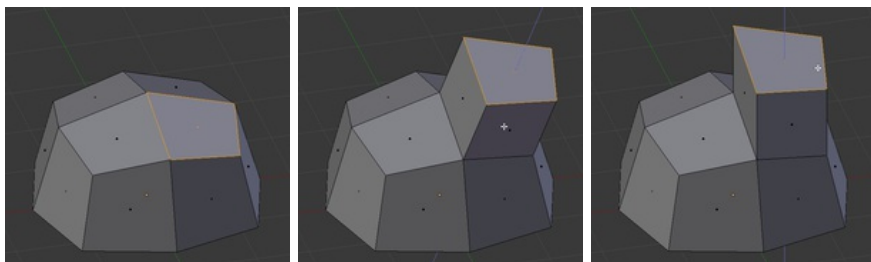
Panel: Mesh Werkzeuge » Extrudieren

Hotkey: E oder AltE

Menu: Mesh » Extrudeire Fläche

Das wichtigste Werkzeug beim Arbeiten eines Meshes ist das Extrudieren. Es erzeugt (standardmäßig) parallele Flächen, Kanten oder Vertices.

Die Auswahl an Vertices wird an ihrer gemeinsamen normalen entlang extrudiert, kann aber auch auf Achsen beschränkt werden.



ausgewählte Fläche

während dem Extrudieren

auf Z begrenzt

Obwohl der Prozess recht intuitiv ist, hier nochmal das zugrundeliegende Prinzip:

- Zuerst findet der Algorithmus die äußere Kante der ausgewählten Fläche um zu ermitteln wo neue Flächen erzeugt werden. Standardmäßig behandelt der Algorithmus Flächeninterne Kanten nicht.
- Diese Kanten werden dann in Flächen verwandelt.
- Falls eine der Kanten nur zu einer Fläche im Mesh angehört, werden alle ausgewählten Flächen dupliziert und an die ursprüngliche Stelle gesetzt. Ein Rechteck wird so zu einem geschlossenen Quader.
- Sonst werden die durch die Kanten neu erzeugten Flächen nur verknüpft.
- Beim extrudieren von geschlossenen Körpern werden diese einfach dupliziert.

individuelle Elemente Extrudieren

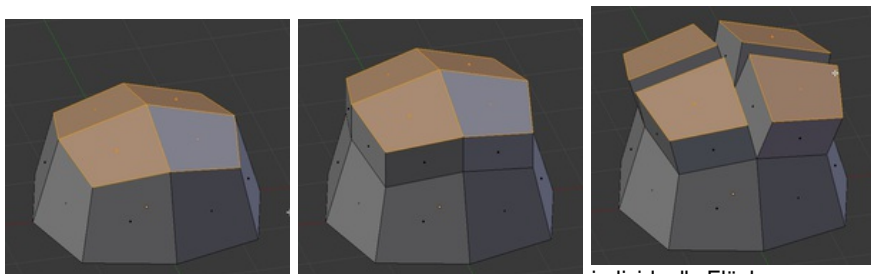
Mode: Bearbeitungsmodus

Panel: Mesh Werkzeuge » Extrudiere individuelle

Hotkey: AltE

Menu: Mesh » Extrudiere individuelle

Extrudiere individuelle erlaubt das extrudieren von individuellen Elementen wie Flächen. Das bedeutet das keine Verallgemeinerungen für das Mesh gemacht werden. Alle ausgewählten Kanten werden als für das Extrudieren der zugehörigen Flächen gültig angesehen und Flächen werden entlang ihrer individuellen Flächennormalen extrudiert.



auswahl mehrerer Flächen

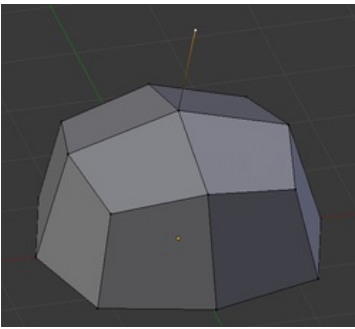
Extrudieren

individuelle Flächen
Extrudieren

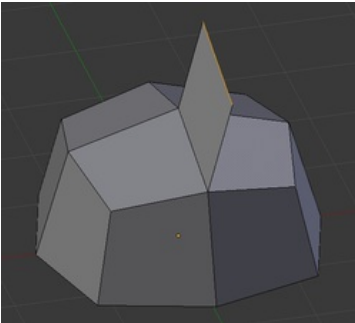
Extrudiere nur Kanten oder Vertices

Mode: Bearbeitungsmodus, Vertex oder Kanten

Hotkey: AltE

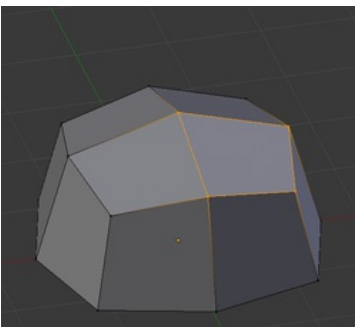


einzelne Vertices extrudieren

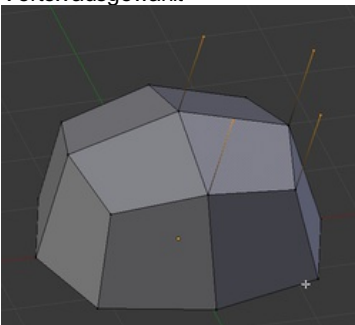


einzelne Flächen extrudieren

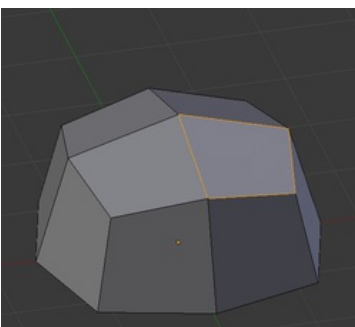
Wenn eine Auswahl eine Kante oder Fläche beschreibt, werden nur Kanten oder Vertices extrudiert. Diese Funktion wird bei ausgewählten Flächen mit AltE aufgerufen.



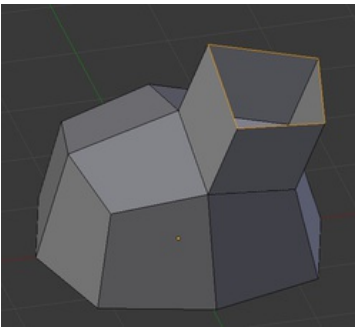
Vertex ausgewählt



nur Vertices extrudieren



Kante ausgewählt



nur Kante extrudieren

Spin (Drehen)

Mode: Bearbeitungsmodus

Panel: Mesh Werkzeuge

Das Spin-Werkzeug erzeugt aus einem Profil ein gedrehtes Objekt in ähnlicher Weise wie eine Drehbank oder Töpferscheibe. Dabei wird die Auswahl um den 3D Cursor als Zentrum und um die Achse in Sichtrichtung gedreht extrudiert.

- Die Sichtachse bestimmt die Rotationsachse.
- Die Position des 3D Cursors bestimmt den Ort der Rotationsachse.

Hier sind die verfügbaren Einstellungen aufgeführt:

Steps (Schritte)

Stellt ein wieviele Unterteilungsschritte beim Extrudieren gemacht werden und hat zur Folge wie rund das Ergebnis wird.

Specifies how many copies will be extruded along the "sweep"

Dupli (Vervielfältigen)

Dupliziert die Elemente, verknüpft sie aber nicht, sodass keine zusammenhängende Fläche entsteht.

Angle (Winkel)

Stellt den Drehwinkel ein.

Center (Zentrum)

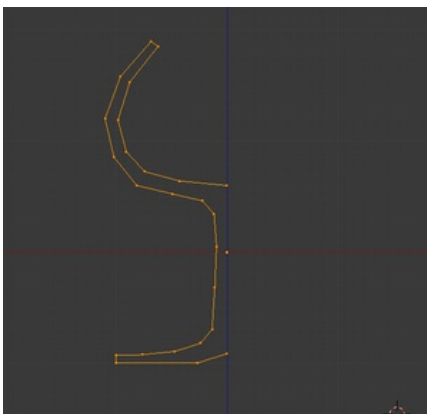
Hier kann das Zentrum des Drehens nachträglich verschoben werden. Standardmäßig ist es die Position des 3D Cursors.

Axis (Achse)

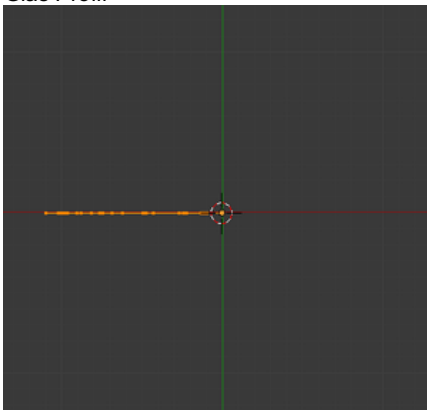
Hier kann nachträglich die Drehachse als Vektor eingestellt werden. Standardmäßig ist es die Sichtachse zum Zeitpunkt des Drehens.

Beispiel

Zuerst muss das Profil des Körpers erstellt werden. Dann muss der 3D Cursor im gewünschten Drehmittelpunkt positioniert werden und in die Draufsicht gewechselt werden. Aktiviere die Funktion im Bearbeitungsmodus mit allen Vertices ausgewählt und stelle den Drehwinkel auf 360°.

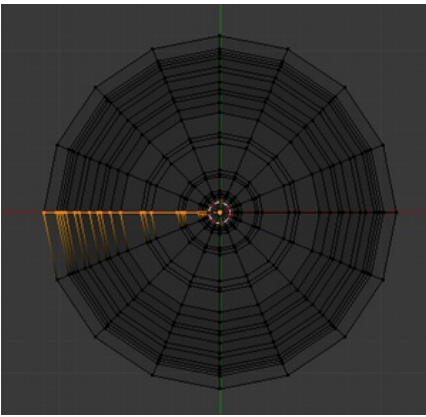


Glas Profil

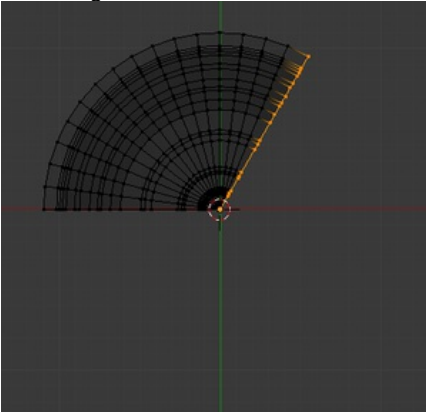


Glas Profil

Winkel

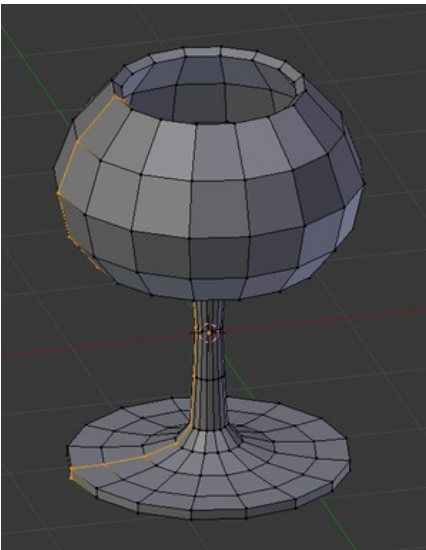


um 360° gedrehtes Profil



um 120° gedrehtes Profil

Dupli

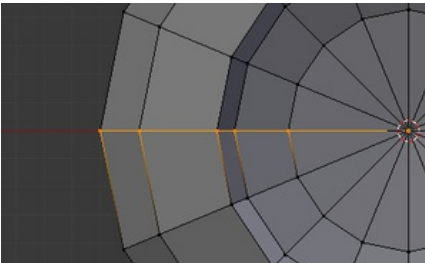


Dupli an



Dupli aus

Verschmelzen oder Entfernen verdoppelter Vertices



Verdoppelte Vertices

Beim Drehen um 360 Grad existieren die erste und die letzte senkrechte Edge Loop auf derselben Position, denn sie werden nicht automatisch verbunden. Es muss also entweder [Merge \(Verschmelzen\)](#) oder [Remove Doubles \(Duplikate entfernen\)](#) angewendet werden.

Normalen Neuberechnen

Es kann sein, dass die Flächennormalen nicht in die gewünschte Richtung zeigen. Dafür sollte die Funktion [Recalculate Normals \(Normalen Neuberechnen\)](#) angewendet werden.

Mesh Unterteilungswerkzeuge

Hier werden die Verschiedenen Unterteilungswerkzeuge aufgeführt.

- [Unterteilung](#)
- [fraktale Unterteilung](#)
- [glatte Unterteilung](#)
- [Schleifen Unterteilung](#)
- [Schneiden Unterteilung](#)
- [Bevel](#)

Unterteilen

Mode: Bearbeitungsmodus

Panel: Mesh Werkzeuge (Bearbeitungskontext)

Hotkey: W » 1 NumPad/2 NumPad

Menu: Mesh » Kanten » Unterteilen, Specials » Unterteilung/weiche Unterteilung

Unterteilen teilt Kanten und Flächen durch halbieren und das Hinzufügen aller notwendigen Vertices.

Optionen

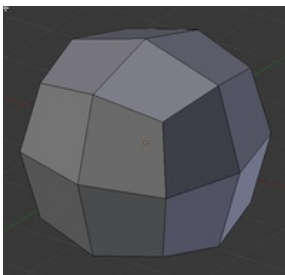
Diese Optionen sind nach ausführen des Werkzeugs verfügbar:

Anzahl an Schnitten

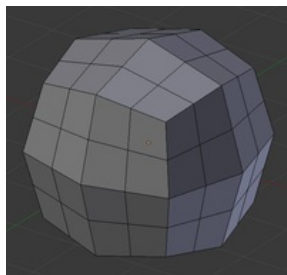
Stellt die Anzahl an Schnitten ein.

Glättung

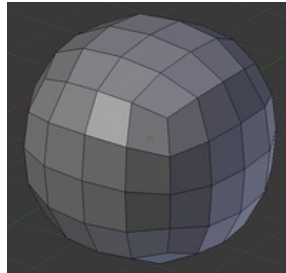
Verschiebt Unterteilungen um das Ergebnis kurvenähnlicher zu machen.



Mesh vor Unterteilung



Unterteilung ohne Glättung



Unterteilung mit einer Glättung von 1

Vier-/Dreiecksmodus

Schaltet um ob eher Dreiecke oder Vierecke erzeugt werden sollen.

Kanten Schnitt Typ

Dieses Menü erlaubt Einstellungen wie Vierecke mit zwei anliegenden Kanten unterteilt werden sollen.

Fächer

Das Viereck wird als Fächer von Dreiecken unterteilt.

Innervert

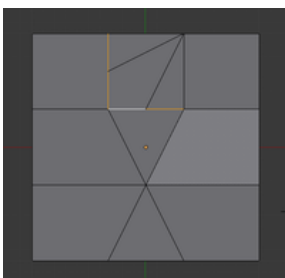
("inneres vertex"), Die ausgewählten Kanten werden unterteilt, dann wird eine Kante zwischen den neu entstandenen Vertices gebildet. Diese neue Kante wird unterteilt und das neu entstandene Vertex wird mit dem nicht ausgewählten Vertex des Vierecks verbunden. Es entstehen also zwei Vierecke und ein Dreieck.

Pfad

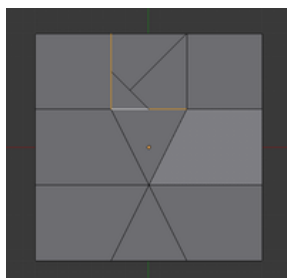
Eine Kante wird zwischen den Mittelpunkten und den Randpunkten der ausgewählten Kanten gebildet.

Straight Cut

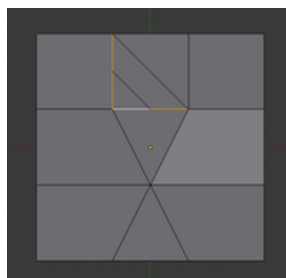
Eine Kante wird zwischen den Mittelpunkten der ausgewählten Kanten gebildet.



Fächer Schnitt Typ



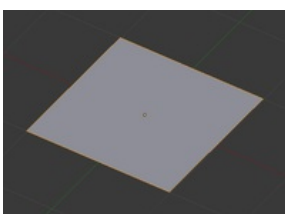
Innervert Schnitt Typ



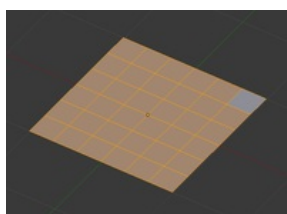
Pfad Schnitt Typ

Fraktal

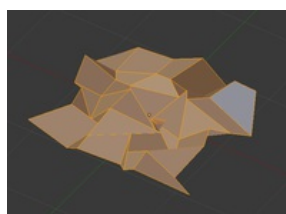
Verschiebt die Vertices zufällig nach der Unterteilung.



Ebene vor Unterteilung



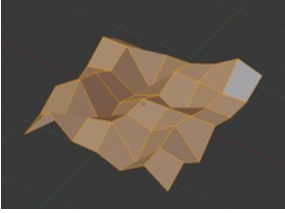
normale Unterteilung



Unterteilung mit Fraktal

Entlang Normalen

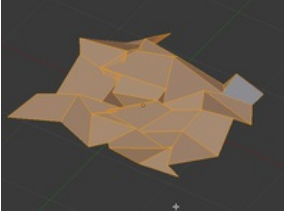
Bewegt Vertices nicht mehr zufällig sondern auf ihren Normalen.



Entlang Normalen auf Stufe 1

Zufallsgeneratorwert

Verändert den Ausgangswert des Zufallsgenerators.



Das selbe Mesh mit unterschiedlichen Zufallswerten.

Schleifenunterteilung

Mode: Bearbeitungsmodus

Panel: Bearbeitungskontext → MeshWerkzeuge

Hotkey: StrgR



Schleifenunterteilung teilt unterteilt eine Flächenschleife mit beliebig vielen Kantenschleifen.

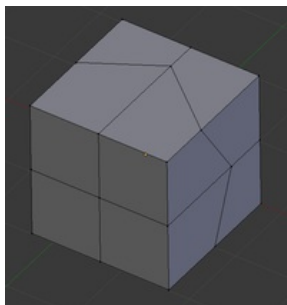
Benutzung

Vorschau der Unterteilung

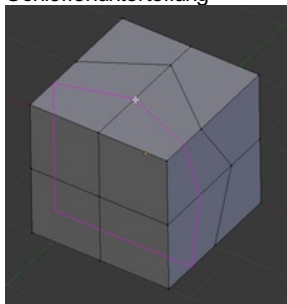
Bei Aktivierung des Werkzeugs muss die Maus über eine Kante bewegt werden. Eine Vorschau für die Kantenschleife die eingefügt würde wird in magenta dargestellt und hört wie Schleifen bei Polen auf.

eine einzelne neue Schleife bewegen

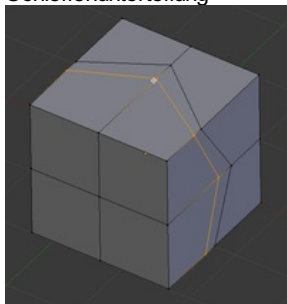
Sobald eine einzelne Kantenschleife mit LMB  platziert ist, kann diese mit der Maus bewegt werden um eine beliebige Position auf der Flächenschleifen einzunehmen und funktioniert genau wie das Werkzeug [Kanten verschieben](#). Entsprechend bestätigt LMB  die Operation an der gewünschten Stelle bricht die Operation ab und setzt die Kante auf die Mitte der Flächenschleife.



Mesh vor der Schleifenunterteilung



Vorschau der Schleifenunterteilung



Interaktive Platzierung der Schleife

Optionen

Während das Werkzeug in Benutzung ist stehen folgende Optionen zur Verfügung:

Gleichförmig E


Nur verfügbar für einzelne Kantenschleifen. Gleicht die Form der neuen Kantenschleife einer benachbarten Kantenschleife an.

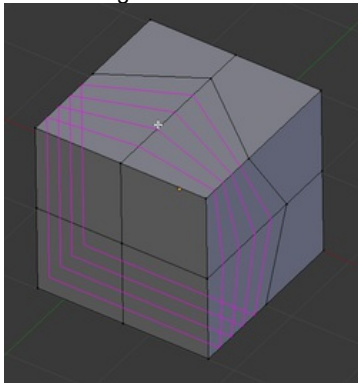
Flip F

Wechselt die Kantenschleife die angenähert wird.

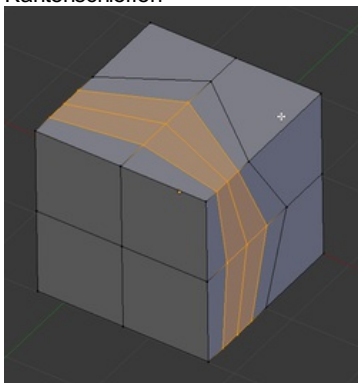
Anzahl

Nach Aktivieren des Werkzeugs aber vor dem ersten Bestätigungsschritt, kann die Anzahl an Kantenschleifen mit dem Mausras

Wheel  oder + NumPad/- NumPad eingestellt werden die erzeugt werden sollen.
 Beim erzeugen von mehreren Schleifen werden diese gleichmäßig über die Flächenschleife verteilt.



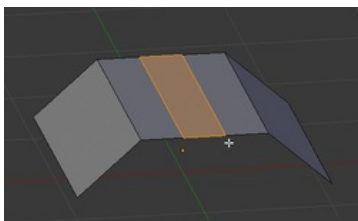
Vorschau für mehrere
Kantenschleifen



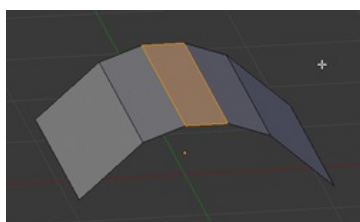
Ergebnis von mehrfacher
Kantenunterteilung

Glätten Alt Wheel

Das Glätten von Kantenunterteilung bewirkt das die Vertices der Kantenschleife bei interpolierten Punkten platziert werden im Gegensatz zum Normalfall wo sie auf den existierenden Kanten platziert werden.



Kantenschleifenunterteilung ohne
Glätten



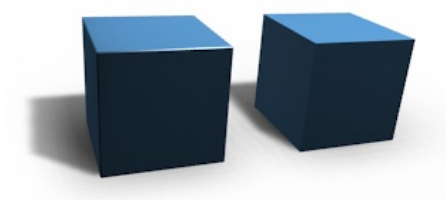
Kantenschleifenunterteilung mit
Glätten

Fase

Modus: Editiermodus

Tastenkombination: StrgB oder W » Fase

Menü: Mesh » Kante » Fase oder StrgE » Fase



Mit und ohne Fase

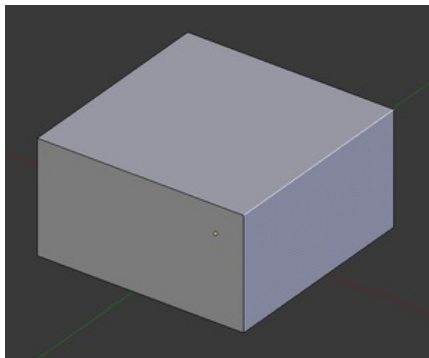
Das Fasen Werkzeug erlaubt das Erstellen von Fasen, abgerundeten und gerade abgeschnittenen Kanten. Diese können wichtig für den Realismus von mechanischen Objekten sein, da diese aus praktischen Gründen in der Realität meistens keine scharfen Kanten haben.

Fasen Modifikator

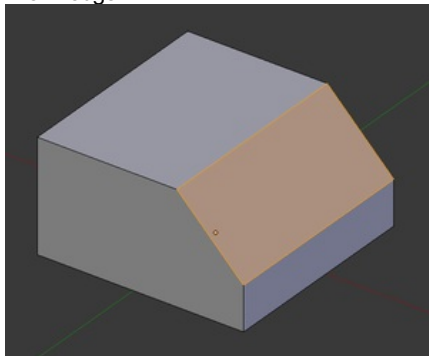
Der [Fasen Modifikator](#) ist eine Alternative zu diesem Werkzeug die das Mesh erst verändert wenn der Modifikator angewendet wird. Es hat die gleichen und ein paar zusätzliche Optionen, wie Kontrolle über die Fase über das Gewicht der Vertices.

Benutzung

Das FasenWerkzeug wird auf alle ausgewählten Kanten angewendet. Fasen können durch Unterteilung abgerundet werden.

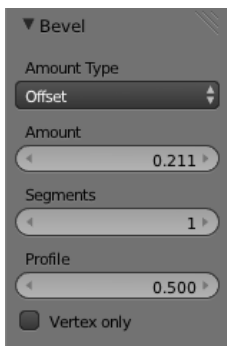


Kante vor Anwendung des Fasen Werkzeugs



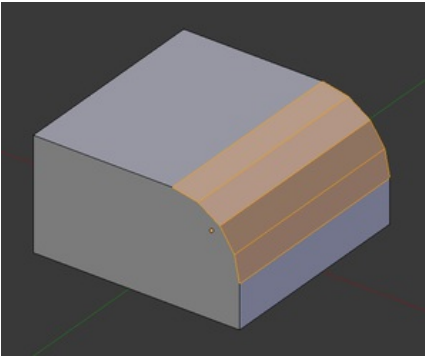
Fase

Optionen



Abstand

Die Breite der Fase kann über den Abstand der entstandenen Fläche zur Kante festgelegt werden. Diese Entfernung kann eingegeben oder mit der Maus eingestellt werden.

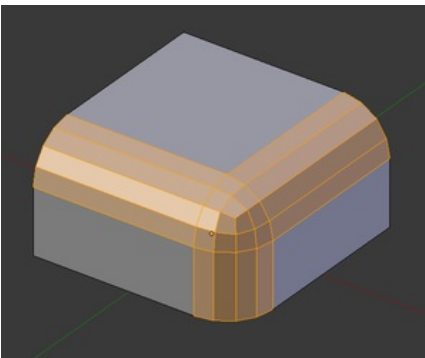


Bevel with 4 segments

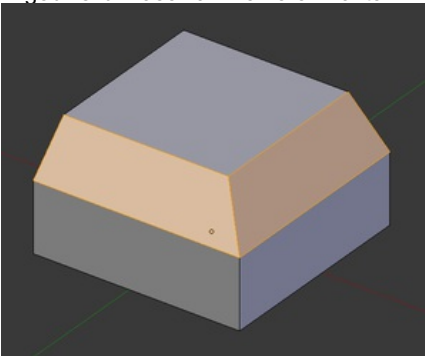
Segmente

Die Anzahl der Segmente der Fase kann ebenfalls eingegeben oder mit dem Mausrad Wheel  eingestellt werden.

Beispiele



Ergebnis für Fasen an mehreren Kanten



Anderes Beispiel für Fasen an mehreren Kanten

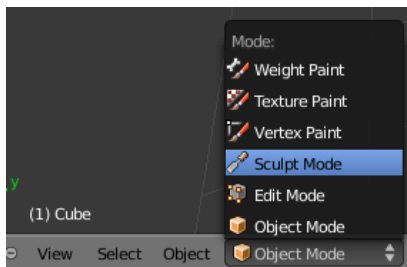
Übersicht

Der Sculpt Mode ähnelt dem Edit Mode dahingehend, dass er zum Bearbeiten der Modellform dient, wenngleich der Sculpt Mode auch eine deutlich abweichende Vorgehensweise verfolgt: Statt auf einzelne Elemente einzuwirken (Vertices, Edges und Faces), wird ein Bereich des Modells mit einem Brush (dt.: Pinsel, Werkzeugspitze) verändert. Mit anderen Worten wählt man im Sculpt Mode automatisch Vertices in Abhängigkeit von der Brush-Position aus und bearbeitet diese entsprechend, ohne dass man die Vertices erst vorher gruppenweise auswählen müsste.

Sculpt Mode

Der Sculpt Mode wird aus der Modusauswahl des Header-Menüs im 3D-Ansichtsfenster (3D View) aufgerufen.

Sobald der Sculpt Mode aktiv ist, werden im seitlichen Toolbar des 3D-Ansichtsfensters (3D View) die modusspezifischen Werkzeugpanels angezeigt: Brush (Pinsel), Texture (Textur), Tool (Werkzeug), Symmetry (Symmetrie), Stroke (Strich), Curve (Kurve), Appearance (Aussehen) und Options (Optionen). Des Weiteren wird ein roter Kreis zu sehen sein, welcher der Cursorposition im 3D-Ansichtsfenster folgt.



Sculpt Mode Dropdown-Menü.

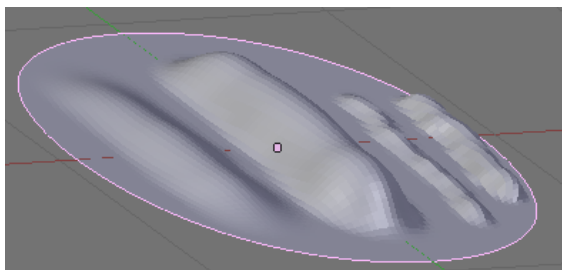


Aussehen des Cursors im Sculpt Mode.

Sculpt-Brushes

Brushes sind Werkzeugvoreinstellungen. Sie sind eine Kombination des jeweiligen Werkzeugs mit Strichart, Textur und weiteren Optionen.

Der Sculpt Mode bietet 16 verschiedene Brushes, die jeder für sich auf eine bestimmte Art auf das Modell einwirken. Bei vielen kann zwischen additiver (hinzufügender) und subtraktiver (verringender) Arbeitsweise hin- und hergeschaltet werden. Sie können aus dem Tool-Menü gewählt werden. Gegenwärtig stehen folgende Brushes zur Verfügung: Blob, Clay, Crease, Draw, Fill/Deepen, Flatten/Contrast, Grab, Inflate/Deflate, Layer, Nudge, Pinch/Magnify, Rotate, Scrape/Peak, Smooth, Snake Hook, Thumb:



Modellieren mit verschiedenen Pinselgrößen und -stärken.

Blob (Tropfen)

Ein- und Ausstülpfen von Rundungen in Meshes.

Clay (Tonerde)

Ahnt dem Draw-Brush, bietet aber zudem die Möglichkeit einzustellen, auf welcher Ebene der Brush wirken soll.

Crease (Knicken)

Erzeugt beim Ziehen oder Drücken des Meshes scharfe Kerben oder Grate, wobei die Vertices zusammengeschoben werden.

Draw (D) (Ziehen)

Bewegt die Vertices ein- oder auswärts auf Grundlage der gemittelten Normalen der Vertices im Bereich des gezogenen Pinselstrichs.

Fill (Auffüllen)

Der Fill-Brush wirkt wie der Flatten-Brush, holt jedoch nur Vertices unterhalb der Brush-Fläche nach oben. Die Umkehrung des Scrape-Brushes dient zum Deepen (Vertiefen) durch Drücken von Vertices unterhalb der Brush-Fläche.

Flatten (T) (Abflachen)

Der Flatten-Brush ermittelt eine 'Bezirksebene', welche standardmäßig auf der Durchschnittshöhe oberhalb bzw. unterhalb der Vertices im Bereich des Pinsels angesiedelt wird. Die Vertices werden dann in Richtung dieser Ebene gezogen. Die Umkehrung des Flatten-Brushs ist der Contrast-Brush, der die Vertices aufwärts oder abwärts von der Pinselebene wegbewegt.

Grab (G) (Greifen)

Grab wird benutzt, um eine Gruppe von Punkten hin- und herzuziehen. Im Unterschied zu den anderen Brushes verändert Grab nicht verschiedene Punkte, wenn der Brush über das Modell gezogen wird. Stattdessen wählt Grab bei gedrückter Maustaste eine Gruppe von Vertices aus und bewegt diese der Maus folgend. Der Effekt ähnelt dem Bewegen einer Gruppe von Vertices im Edit Mode bei eingeschaltetem Proportional Edit; allerdings kann man mit Grab auch noch andere Optionen des Sculpt Modes (wie Texturen und Symmetrie) einbeziehen.

Inflate (I) (Aufblasen)

Ähnelt Draw, jedoch mit dem Unterschied, dass Vertices im Inflate-Modus in die Richtung ihrer jeweiligen eigenen Normalen verschoben werden.

Layer (L) (Schichten)

Dieser Pinsel ähnelt Draw, unterscheidet sich allerdings dadurch, dass die Höhe der Schicht während des Auftrags auf einen fixen Wert gesetzt wird. Dadurch entsteht der Eindruck, dass eine feste Schicht aufgetragen wird. Man kann mit diesem Brush auch nicht übereinandermalen, sondern Pinselstriche durchdringen einander. Wenn man jedoch die Maustaste loslässt und einen neuen Pinselstrich beginnt, wird die Schichtdicke zurückgesetzt und man kann auch einen vorherigen Strich übermalen.

Nudge (Anstupsen)

Bewegt Vertices in die Richtung des Pinselstrichs.

Pinch (P) (Zusammenziehen)

Pinch zieht Vertices zum Mittelpunkt des Pinsels hin zusammen. Die gegenteilige Einstellung Magnify, bei der Vertices vom Mittelpunkt des Pinsels aus fortbewegt werden.

Rotate (Rotieren)

Rotiert Vertices im Einflussbereich des Brushes in die Richtung, in die der Cursor bewegt wird.

Scrape (Schaben)

Der Scrape-Brush arbeitet wie der Flatten-Brush, zieht aber lediglich Vertices oberhalb der Ebene abwärts. Die Umkehrung von Scrape ist Peak (Spitzen erzeugen) durch Ziehen von Vertices oberhalb und fort von der Ebene.

Smooth (S) (Glätten)

Wie der Name schon vermuten lässt, werden im Einflussbereich des Pinsel Unregelmäßigkeiten auf dem Mesh durch Nivellierung der Vertexpositionen beseitigt.

Snake Hook (Schlangenlinie)

Zieht Vertices entlang der Pinselbewegungen und erzeugt langgezogene schlangenlinienartige Formen.

Thumb (Abgreifen)

Ähnelt dem Nudge-Brush, wobei dieser das Mesh im Einflussbereich des Pinsels bei gleichzeitigem Bewegen in Richtung des Pinselstrichs glättet.

Sculpten mit dem Multires Modifier

...

Sculpt Properties Panel

Dieses Panel erscheint in der Werkzeugpalette an der linken Seite des 3D-Ansichtsfensters.

Brush Menu (Pinsel-Menü)

Radius (Pinselgröße)

Diese Option steuert den Radius des Brushes in Pixeln gemessen. Das Tastenkürzel F im 3D-Ansichtsfenster erlaubt das interaktive Ändern der Pinselgröße durch Ziehen der Maus mit abschließendem Linksklick (Die Textur des Pinsels sollte innerhalb des Kreises zu sehen sein). Wenn man F gedrückt hat, anschließend eine Zahl eintippt und dann die Eingabetaste betätigt, kann man die Größe auf einen numerischen Wert festlegen. Die Pinselgröße kann auch durch Einschalten des Icons für die Drucksensitivität beeinflusst werden, sofern ein Tablet benutzt wird, dass diese Funktion unterstützt.

Strength (Stärke)

Strength steuert, in welchem Maße jede Anwendung des Brushes das Modell beeinflusst. Beispielsweise bewirken höhere Werte beim Draw-Brush, dass dem Modell viel schneller Tiefe hinzugefügt wird, während beim Smooth-Brush das Glätten rascher vonstatten geht. Diese Einstellung ist nicht für die Brushes Grab, Snake Hook und Rotate verfügbar.

Wenn der Umfang der zur Verfügung stehenden Stärkeeinstellungen nicht mit dem Modell übereinzustimmen scheint (z.B. wenn sogar die unterste Stärkeeinstellung viel zu starke Veränderungen auf einmal am Modell bewirkt), dann sollte man das Modell insgesamt skalieren (im Edit Mode, nicht im Object Mode). Größere Abmessungen verringern die Pinselwirkung und umgekehrt. Man kann die Pinselstärke interaktiv durch Drücken der Taste \diamond ShiftF im 3D-Ansichtsfenster mit abschließendem Linksklick verändern. Alternativ kann man nach Betätigen von \diamond ShiftF auch einen Zahlenwert für die Stärke eingeben. Die Stärke kann auch durch Einschalten des Icons für die Drucksensitivität beeinflusst werden, sofern ein Tablet benutzt wird, das diese Funktion unterstützt.

Autosmooth (Automatisches Glätten)

Festsetzen des Wertes für das Glätten, das bei jedem Pinselstrich angewendet werden soll.

Sculpt Plane (Sculptebene)

Man benutzt dieses Menü, um die Ebene festzulegen, auf der das Sculpten stattfinden soll.

Plane Offset (Ebenenversatz)

Einrichten der Ebene, auf welcher der Brush zum Betrachter hin oder von ihm weg wirksam werden soll.

Trim (Justierung)

Ermöglicht die Justierung der Sculptebene in Abhängigkeit von der Distance-Einstellung.

Front Faces Only (Nur Frontseiten)

Wenn aktiviert, wird der Pinsel nur die Vertices beeinflussen, die sich frontal zum Betrachter befinden.

Accumulate (Übereinanderlegen)

Bewirkt, dass Pinseltupfen sich übereinanderlegen.

Stroke Menu (Stricharten-Menü)**Stroke Method** (Strichanwendung)

Legt die Art und Weise fest, wie Pinselstriche auf dem Mesh aufgetragen werden:

Dots (Tupfen)

Standard-Pinselstrich.

Drag Dot (gezogenes Tupfen)

Erzeugt einen einzelnen Versatz in der Pinselform. Erst klicken, dann auf dem Mesh an die gewünschte Stelle ziehen und schließlich loslassen.

Space (Abstand)

Erzeugt einen Pinselstrich als eine Abfolge von Tupfen, deren Abstände durch die Spacing-Einstellung vorgegeben werden. Spacing steht dabei für den Prozentwert des Pinseldurchmessers.

Anchored (Verankert)

Erzeugt einen einzelnen Versatz an der Position des Pinsels. Durch Klicken und Ziehen wird der Pinseldurchmesser verändert. Bei Edge to Edge (Kante zu Kante) wird die Pinselposition und -ausrichtung durch einen Zwei-Punkt-Kreis bestimmt, wobei der erste Klick einen Ausgangspunkt erzeugt und das Ziehen den zweiten Punkt gegenüber dem ersten platziert.

Airbrush (Spritzpistole)

Der Pinselfluss hält solange an, wie die Maus gedrückt gehalten wird und bestimmt durch die Rate-Einstellung. Wenn diese deaktiviert ist, verändert der Pinsel das Modell nur, wenn er seine Position verändert. Diese Option ist nicht für den Grab-Pinsel verfügbar.

Die folgenden Parameter sind verfügbar für die Stricharten Dots, Space und Airbrush:

Smooth stroke (Weicher Pinselstrich)

Der Pinsel folgt der Maus verlangsamt und bewegt sich auf einem weichen Pfad. Sofern eingeschaltet, werden die folgenden Einstellungen aktiv:

Radius (Radius)

Festlegen des minimalen Abstands vom letzten Punkt, bevor der Pinselstrich fortgesetzt wird.

Factor (Faktor)

Festlegen des Weichheitsgrads.

Jitter (Zittern)

Zitternde Pinselführung während des Malens.

Curve Menu (Kurven-Menü)

Die Curve-Sektion erlaubt es, über eine Kurve auf der rechten Seite die Intensität des Pinsels von dessen Mitte ausgehend (linker Teil der Kurve) hin zu dessen Rändern (rechter Teil der Kurve) einzustellen.

Texture Menu (Texturen-Menü)

Auch eine Textur kann dazu verwendet werden, um die Stärke von Pinseleffekten festzulegen. Hierzu wählt man entweder eine bereits vorhandene Textur aus der Texturauswahl oder man erstellt eine neue durch Betätigen der New-Taste.

Brush Mapping (Pinsel-Mapping)

Legt die Art und Weise fest, wie eine Textur auf den Pinselstrich gemappt (aufgebracht) wird:

Fixed (Fixiert)

Ist Fixed eingeschaltet, folgt die Textur der Maus, so dass es so aussieht, als wenn die Textur über das Modell gezogen würde.

Tiled (Gekachelt)

Die Tile-Option kachelt eine Textur über den Schirm, so dass die Pinselbewegungen so wirken, als ob sie sich von der Textur getrennt bewegen würden. Die Tile-Funktion ist am nützlichsten mit Bildern, die gekachelt werden können, weniger mit prozeduralen Texturen.

3D (Dreidimensional)

Die 3D-Option ermöglicht es, mit dem Pinsel die Vorteile prozeduraler Texturen voll auszuschöpfen. Dieser Modus benutzt Vertex-Koordinaten statt der Pinselposition, um festzulegen, welcher Bereich der Textur verwendet werden soll.

Angle (Winkel)

Gemeint ist der Rotationswinkel des Texturenpinsels. Er kann interaktiv geändert werden durch die Tastenkombination StrgF im 3D-Ansichtsfenster. Befindet man sich in der interaktiven Rotation, kann auch ein numerischer Wert eingegeben werden. Dieser kann gesetzt werden auf:

User (Benutzer)

Direkte Eingabe der Gradzahl.

Rake (Neigungswinkel)

Der Winkel folgt der Richtung des Pinselstrichs. Nicht verfügbar für 3D-Texturen.

Random (Zufallswert)

Zufällige Gradzahl.

Offset (Versatz)

Feineinstellung der Texturenplatzierung auf der X-, Y- und Z-Achse.

Size (Größe)

Diese Einstellung erlaubt die Änderung des Vergrößerungsfaktors der Textur. Nicht verfügbar für Drag-Texturen.

Sample Bias (Probenverzerrung)

Zu Texturenproben hinzuaddierter Wert.

Overlay (Überlagerung)

Wenn eingeschaltet, wird die Textur im 3D-Ansichtsfenster so dargestellt wie vorgegeben durch den

Alpha (Transparenz)-Wert.**Symmetry Menu (Symmetrie-Menü)**

Spiegelt die Pinselstriche an der ausgewählten lokalen Achse. Bitte beachten, dass wenn man die Achsenrichtung ändern möchte, das Modell im Edit Mode, nicht im Object Mode gedreht werden muss.

Feather (Feder)

Verringert die Strichstärke an Stellen, an denen sich die Symmetrieebenen überschneiden.

Radial (Radial)

Diese Einstellungen ermöglichen radiale Symmetrie auf den gewünschten Achsen. Die Zahl bestimmt, wie oft der Strich innerhalb von 360° um die zentrale Achse wiederholt werden soll.

Options Menu (Optionen-Menü)**Threaded Sculpt** (Gewundenes Modellieren)

Bedient sich der Vorteile von Mehrfach-CPU-Prozessoren, um die Modellierungsfähigkeiten zu verbessern.

Fast Navigation (Schnelles Navigieren)

Zeigt bei Multires-Modellen einer niedrigere Auflösung während der Navigation im 3D-Ansichtsfenster an.

Show Brush (Pinselanzeige)

Anzeige der Pinselform im 3D-Ansichtsfenster.

Gemeinsame Einstellungen

; **Size** (Größe)

Festsetzen der gleichen Pinselgröße für alle zur Verfügung stehenden Pinsel.

; **Strength** (Stärke)

Festsetzen des gleichen Stärke des Pinselauftrags für alle zur Verfügung stehenden Pinsel.

Lock (Sperren)

Diese drei Tasten ermöglichen es, während des Modellierens alle Veränderungen/Verformungen des Modells auf ausgewählte Achsen zu begrenzen.

Appearance Menu (Darstellungsmenü)


Man kann die Pinselfarbe dahingehend festlegen, ob er sich im additiven oder subtraktiven Modus befindet.

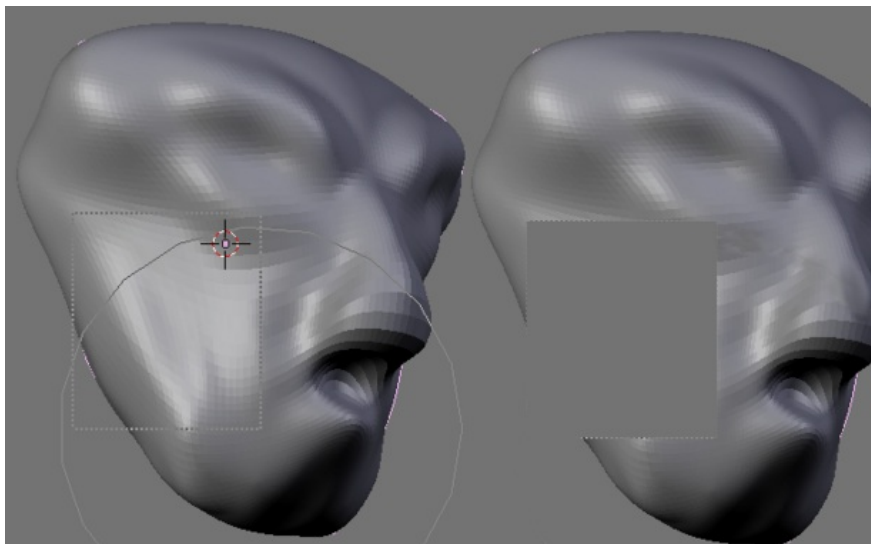
Des weiteren könnte man auch ein Pinselsymbol aus einer Bilddatei laden.

Tool Menu (Werkzeugmenü)

Hier kann man die Einstellungen für einen zu benutzenden Pinsel festlegen. Reset Brush setzt die Einstellungen eines Pinsels auf seine Standardwerte zurück. Man kann Blender auch so einstellen, dass der gegenwärtig aktive Pinsel für den Vertex Paint Mode und den Texture Paint Mode benutzt werden kann, indem man die Umschalttasten benutzt.

Ausblenden und Hervorholen von Mesh-Teilen

Manchmal ist es nützlich, wenn man beim Modellieren Teile eines Meshes getrennt bearbeiten kann. Um einen Teil des Meshes auszublenden, drückt man H und zieht daraufhin mit gedrückter Taste LMB  einen Rahmen um das Teil, das man unsichtbar machen möchte. Um alle ausgeblendeten Teile wieder sichtbar zu machen, drückt man die Tastenkombination AltH.




Vor und nach dem Ausblenden.

Tastenkürzel

Bedienung

Ausblenden von Mesh-Teilen innerhalb der Auswahl
 Gesamtes Mesh wieder sichtbar machen
 Aufruf Stroke-Dialogbox
 Interaktives Einstellen der Pinselgröße
 Interaktives Einstellen der Stärke des Pinselauftrags
 Interaktives Rotieren der Pinseltextur
 Draw-Pinsel
 Smooth-Pinsel
 Pinch-Pinsel
 Inflate-Pinsel
 Grab-Pinsel
 Layer-Pinsel
 Flatten-Pinsel
 X-Symmetrie
 Y-Symmetrie
 Z-Symmetrie
 Einen Multires-Level mehr einstellen
 Einen Multires-Level weniger einstellen

Tastenkürzel

H LMB 
 AltH
 A
 F
 ⇧ ShiftF
 CtrlF
 D
 S
 P
 I
 G
 L
 T
 X
 Y
 Z
 Page up
 Page down

Vertexgruppen (Vertex Groups)

Ein Mesh ist eine Menge verbundener Vertices; in komplexeren Objekten sind es oft tausende. Blender ermöglicht es deshalb, Vertices zu Gruppen zusammenzufassen und gemeinsam auszuwählen. Dieselben Vertices können dabei auch mehreren Gruppen gleichzeitig zugeordnet sein. Dies geschieht zu verschiedenen Zwecken:

- Wiederverwenden von Teilen eines Objekts durch Erstellen von Kopien,
- Verstecken von Vertices, die beim Arbeiten an Details stören würden,
- Dokumentieren und Erklären von Objektteilen für andere,
- Verbiegen von Armatures (Skeletten),
- Erzeugen von Partikeln für die Vertexgruppe,
- Kontrollieren der Geschwindigkeit der emittierten Partikel,
- Hinzufügen mehrerer Materialien zu einem Objekt,
- Steuern von Modifiern oder Constraints (Beschränkungen).

Skelette

Vertexgruppen können automatisch für alle Bones (Knochen) in einem Skelett erstellt werden. Siehe dazu [hier](#). Dieses Kapitel konzentriert sich ausschließlich auf benutzerdefinierte Vertexgruppen.

Wozu Vertexgruppen verwenden?

Vertexgruppen kennzeichnen Untermengen eines Objekts wie zum Beispiel die Beine eines Stuhls oder die Angeln einer Tür. Indem solche Bereiche in Vertexgruppen zusammengefasst werden, kann man diese einfach auswählen und mit ihnen gezielt arbeiten. Mit der Hide-Funktion kann man alles andere aus der Ansicht ausblenden.

Vertexgruppen vereinfachen es, Teile eines Meshes abzutrennen und zu vervielfältigen - man denke zum Beispiel an die Knöpfe eines Legosteins. Der einfachste Stein besteht aus einer Basis und einem Knopf. Um einen vierknöpfigen Stein zu bauen, wählt man die Knopf-Vertices aus, dupliziert sie im Edit Mode und positioniert sie dort, wo benötigt werden.

Eine andere Verwendung für Vertexgruppen ist das Verhüllen eines Skeletts. Wenn man das Mesh animieren möchte, definiert man ein Skelett aus unsichtbaren Bones. Bewegt sich ein Bone, dann folgen die mit ihm verbundenen Vertices der Bewegung, aber nur diejenigen, die ihm zugeordnet sind. Wenn man also einen Arm-Bone bewegt, dann bewegt dieser die Arm-Vertices und nicht die Bein-Vertices. So werden Teile des Meshs bewegt und gestreckt, während andere unbeeinflusst bleiben.

Das [Partikelsystem](#)-Menü hat ein Vertexgruppen-Panel, in dem verschiedene Eigenschaften auf bestimmte Vertexgruppen beschränkt werden können. Das Vertexgruppen-Weightpainting steuert die Menge, Größe und Geschwindigkeit von Partikeln. Man kann zum Beispiel bei einem Partikelsystem für [Haare](#) eine Vertexgruppe mit Namen "Scalp" benutzen, um Haare nur von einem Teil des Kopfes ausgehen zu lassen.

Außerdem nutzen viele Modifikatoren die [Vertexgruppen](#), um ihren Einfluss abzugrenzen. Einige Modifikatoren wie der [Maskenmodifikator](#) brauchen eine Vertexgruppe, um überhaupt arbeiten zu können.

Note

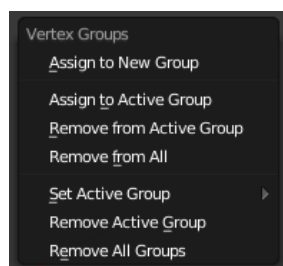
Vertexgruppen sind nicht immer nötig, um Eigenschaften einer begrenzten Auswahl von Vertices zuzuweisen. Wenn zum Beispiel mehrere Materialien einem Mesh hinzugefügt werden, dann müssen diese einem Satz von Vertices zugewiesen werden. Es kann aber darauf verzichtet werden, die Vertices dafür einer Vertexgruppe zuzuweisen, denn die Auswahl kann später auch wieder über das Material erfolgen.

Das Vertexgruppen-Menü

Im Bearbeitungsmodus werden mit StrgG Vertexgruppen verwaltet.

Existiert noch keine Vertexgruppe, so wird nach vorheriger Auswahl von Vertices eine neue Vertexgruppe mit Namen "Group" für die ausgewählten Vertices erzeugt.

Existiert bereits eine Vertexgruppe, so wird das Vertexgruppen-Menü geöffnet. Dieses bietet folgende Optionen:



Vertex Groups Popup-Menü

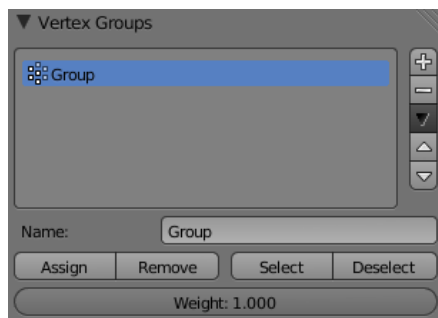
- Zuweisen der ausgewählten Vertices zu einer neuen Vertexgruppe
- Zuweisen der ausgewählten Vertices zur aktiven Gruppe
- Entfernen der ausgewählten Vertices aus der aktiven Gruppe
- Entfernen der ausgewählten Vertices aus allen Gruppen

- Festlegen der aktiven Gruppe
- Entfernen der aktiven Gruppe
- Entfernen aller Gruppen

Vertexgruppen gibt es nur für Meshes

Vertexgruppen gibt es nur für Objekte, die Vertices besitzen. Textobjekte zum Beispiel können keine Vertexgruppen haben und das Menü wird nicht geöffnet, wenn ein solches Element ausgewählt ist.

Erzeugen und Löschen



Vertex Groups Panel im Edit Mode

Standardmäßig besitzt ein Objekt keine Vertexgruppen und alle Vertices sind ungruppiert. In manchen Zusammenhängen können Vertexgruppen automatisch generiert werden, zum Beispiel beim Weightpainting. Normalerweise erzeugt und ändert man Vertexgruppen jedoch über das Vertex group panel im Object data Menü. Sobald eine Vertexgruppe hinzugefügt worden ist und man im Edit Mode ist, wird das Vertexgruppenmenü zu einem Tastenfeld.

Dies kann durch die erste Option des Vertexgruppen-Menüs oder durch manuelles Erstellen einer Gruppe im Vertexgruppen-Bereich des Eigenschaftensfensters erreicht werden. Allerdings wird dort nur eine neue Gruppe erzeugt; Vertices werden der Gruppe nicht automatisch zugeordnet. In diesem Menü können Vertexgruppen außerdem umbenannt werden.

Überprüfe die Zuweisung

Es ist eine gute Idee, durch die Select- und Deselect-Tasten die korrekte Zuweisung von Vertices zu überprüfen. Erforderlichenfalls drückt man die Assign-Taste, um ausgewählte Vertices zur Vertexgruppe nachträglich hinzuzufügen.

Um eine Vertexgruppe zu löschen, wählt man sie in der Liste aus und klickt auf die Taste -. Alle der Gruppe zugewiesenen Vertices werden aus der Gruppe entfernt. Bitte beachten, dass ein Vertex zu mehreren Gruppen gehören kann! Sie sind dann weiterhin Teil der anderen Gruppen.

Hinzufügen und Entfernen von Vertices

Um Vertices zu einer bestehenden Vertexgruppe hinzuzufügen, geht man wie folgt vor:

1. Auswahl der Vertexgruppe, mit der man arbeiten möchte.
2. Man markiert mit **⇧ Shift RMB** oder **Strg LMB** weitere Vertices für die Vertexgruppe aus.
3. Betätigen der Taste Assign oder Aufruf der Option **StrgG » Add Selected to Active Group**.

Bitte nicht vergessen, dass ein Vertex auch mehreren Gruppen zugeordnet sein kann.

Note

Die Taste Assign fügt nur die ausgewählten Vertices zur aktiven Vertexgruppe hinzu. Bereits zugeordnete Vertices werden nicht aus der Gruppe entfernt.

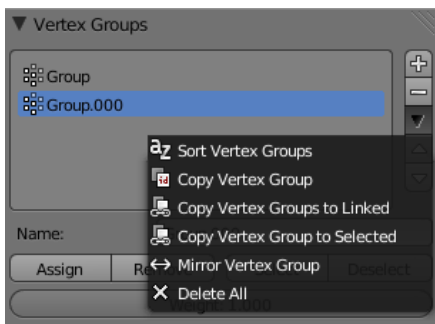
Vertices aus der Gruppe entfernt man wie folgt:

1. Auswahl der Vertexgruppe, mit der man arbeiten möchte.
2. Man markiert mit **⇧ Shift RMB** oder **Strg LMB** Vertices aus, die aus der aktiven Vertexgruppe entfernt werden sollen.
3. Betätigen der Taste Remove oder Aufruf der Option **StrgG » Remove from Active Group**.

Note

Man kann ausgewählte Vertices aus allen Gruppen auf einmal entfernen, indem man im Vertex Groups Menü (**StrgG » Remove from All**) auswählt.

Vertexgruppen-Management



Optionen im Vertex Groups Panel

Sortiere Vertexgruppen

Sortiert Vertexgruppen alphabetisch.

Kopiere Vertexgruppe

Erzeugt eine Kopie der aktiven Vertexgruppe (kopiert keine Vertices!).

Kopiere Vertexgruppe zu Verknüpften

Kopiert Vertexgruppe zu allen verknüpften Duplikaten.

Kopiere Vertexgruppe zu Ausgewählten

Kopiert Vertexgruppe zu allen ausgewählten Meshes.

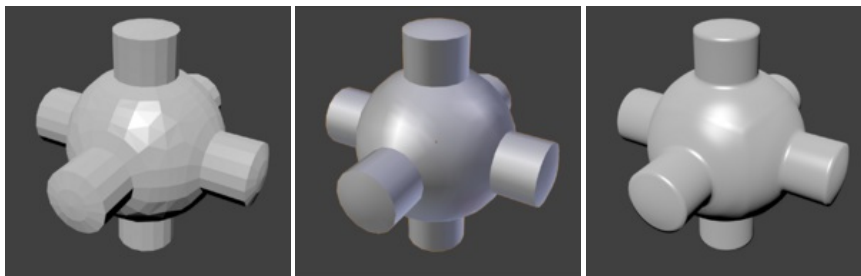
Spiegele Vertexgruppe

Spiegelt die Vertexgruppe und weist einen entsprechenden Namen zu.

Lösche Alle

Löscht alle Vertexgruppen.

Weiche Meshes



Beispiel Mesh, flach gerendert mit weichen Kanten Spaltung, und mittels einer Unterteilung der Oberfläche. Achten Sie darauf wie die Kanten verschieden gerendert werden. [Beispiel_blend](#)

Wie in den vorherigen Kapiteln angesprochen, dreht es sich in Blender zentral um Polygone. Die meisten Objekte werden mit Polygonen dargestellt und echte Kurven-Objekte nähern sich an Polygon Meshes an. Beim Rendern von Bildern, bemerken Sie, dass Polygone als Reihe von kleinen, flachen Flächen erscheinen.

Manchmal ist das ein gewünschter Effekt, aber für gewöhnlich möchten wir das unsere Objekte glatt und weich erscheinen. Dieses Kapitel erläutert wie Objekte weich dargestellt werden können, und wie der Filter Auto Weichzeichnen angewandt werden kann um schnell und einfach weiche und flache Polygone im selben Objekt zu kombinieren.

Die letzte Abteilung dieser Seite zeigt die Möglichkeiten auf um die Geometrie eines Objektes glatt und weich erscheinen zu lassen, nicht nur das Erscheinungsbild.

Weiches Shading

Modus: Mesh und Objekt Modus

Panel: Mesh-Werkzeuge (Mesh Kontext)

Tastenkombination: StrgF » Weich / Flach schattieren

Menü: Mesh » Flächen » Weich / Flach schattieren

Der einfachste Weg ein gesamtes Objekt als weich oder flach anzuzeigen, ist durch das Auswählen des Mesh-Objektes, und im Objekt Modus auf Weich in der Tool Shelf zu klicken. This button does not stay pressed, it forces the assignment of the “smoothing” attribute to each face in the mesh, also when you add or delete geometry.

Beachten Sie dass der Umriss des Objektes weiterhin stark flach erscheint. Das Aktivieren von Funktionen zur Schattierung hat nichts mit der tatsächlichen Geometrie des Objektes gemeinsam; es verändert die Weise auf die das Shading zwischen den Oberflächen berechnet wird, was die Illusion einer weichen Oberfläche vermittelt. Das Drücken vom Knopf Flach in der Abteilung Shading panel in der Werkzeugleiste setzt das Shading wider zurück, wie im ersten oberen Bild dargestellt.



Selbe weiche Schattierung

Teile eines Mesh weich schattieren

Alternativ können Sie wählen welche Kanten weich schattiert werden sollen durch das Betreten des Editiermodus, die Auswahl einiger Flächen und das Klicken auf den Knopf Weich. Die ausgewählten Kanten erscheinen in Gelb.

Wenn sich das Mesh im Editiermodus befindet, only the selected edges will receive the “smoothing” attribute. You can set edges as flat (removing the “smoothing” attribute) in the same way by selecting edges and clicking the Flach button.

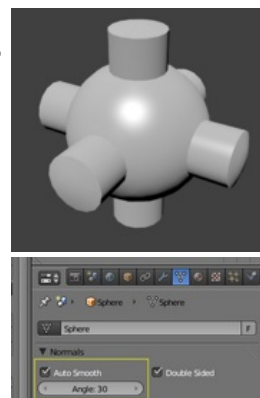
Auto Weichzeichnen

Panel: Eigenschaften (Objektdaten Kontext)

It can be difficult to create certain combinations of smooth and solid faces using the above techniques alone. Though there are workarounds (such as splitting off sets of faces by selecting them and pressing Y), there is an easier way to combine smooth and solid faces, by using Auto Weichzeichnen.

Auto Weichzeichnen can be enabled in the mesh's panel in the Properties window. Angles on the model that are smaller than the angle specified in the Angle button will be smoothed during rendering (i.e. not in the 3D view) when that part of the mesh is set to smooth. Higher values will produce smoother faces, while the lowest setting will look identical to a mesh that has been set completely solid.

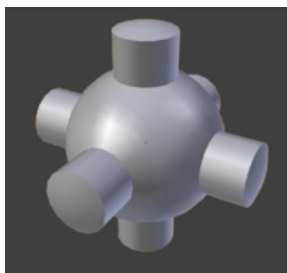
Note that a mesh, or any faces that have been set as Flach, will not change their shading when Auto Weichzeichnen is activated: this allows you extra control over which faces will be smoothed and which ones won't by overriding the decisions made by the Auto Weichzeichnen algorithm.



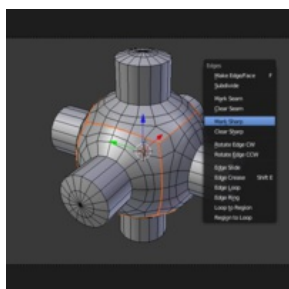
Beispiel Mesh mit

Kanten glätten Modifikator

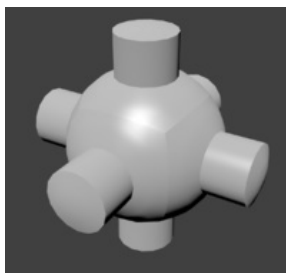
Mit dem Modifikator [Kanten teilen](#) erzielen wir ähnliche Ergebnisse zu Auto Weichzeichnen mit der Möglichkeit zu wählen welche Kanten genau geglättet werden sollen, basierend auf scharf markierten Winkeln.



Aktivierter Kanten glätten Modifikator, basierend auf Winkeln



Scharf markierte Kanten



Resultierender Render mit Gewichtung nach scharfen Kanten

Glätten der Mesh Geometrie

Die oberen Techniken verändern nicht das Mesh selber, sondern nur die Weise auf welche diese dargestellt und gerendert werden. Anstatt das Mesh wie eine weiche Oberfläche erscheinen zu lassen, können Sie auch physikalisch weiche Schattierungen der Geometrie des Meshes mit folgenden Werkzeugen vornehmen:

Werkzeuge zur Mesh-Bearbeitung

Sie können eine der folgenden im Editiermodus anwenden:

[Smooth](#)

This relaxes selected components, resulting in a smoother mesh

[Laplacian Smooth](#)

Glättet die Geometrie eines Smooths geometry by offers controls for better preserving larger details

[Subdivide Smooth](#)

Adjusting the smooth parameter after using the subdivide tool results in a more organic shape. This is similar to using the subdivide modifier.

[Bevel](#)

Die ausgewählten Kanten der Fasen, bewirkt das scharfte Kanten flach dargestellt werden.

Modifikatoren

Alternativ können sie das Mesh auch non-destruktiv mit einer oder mehreren der folgenden Modifikatoren weich schattieren:

[Weicher Modifikator](#)

Works like the Smooth tool in Editiermodus; can be applied to specific parts of the mesh using vertex groups.

[Weicher Modifikator](#)

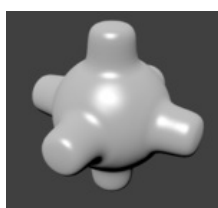
Works like the Laplacian Smooth tool in Editiermodus; can be applied to specific parts of the mesh using vertex groups.

[Fasen Modifikator](#)

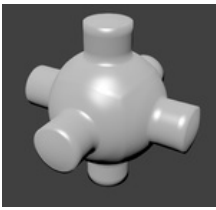
Works like the Bevel tool in Editiermodus; Fasen kann gesetzt werden um mit Winkel Schwellen, oder Gewichtungswerten von Kanten zu arbeiten.

[Unterteilungsoberflächen Modifikator](#)

Catmull-Clark Unterteilung produziert weiche Ergebnisse. Scharfe Kanten können mit [Unterteilungs-Falten](#) definiert werden oder durch das Setzen von bestimmten Kanten zu "scharf" und das Hinzufügen des [Kantenteilen Modifikators](#) (gesetzt auf From Marked Als Scharf, fuzzy) vor dem Oberflächenunterteilung Modifikator.



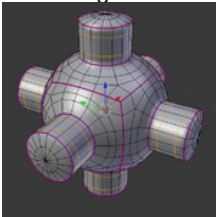
Subsurf



Mit Kanten Falten,
und resultierenden
Unterteilungs
Artefakten



Extra hinzugefügte
Kantenringe



3D Ansicht mit
Kanten Falten (pink)
und hinzugefügten
Kantenringe

Text Objekte

Mode: Bearbeitungsmodus

Menu: Hinzufügen » Text



Text Beispiele

Text Objekte sind ganz einfach Objekte die Text darstellen und erzeugen können. Sie haben ähnliche Eigenschaften wie Kurven und Oberflächen da moderne Schriftarten wie OpenType, TrueType, etc. vector basiert sind und aus Kurven bestehen (normalerweise Beziern).

Blender benutzt ein Font System um zu organisieren wie Schriften im 3D Raum dargestellt werden. Es ist möglich Schriften zu importieren (PostScript Type 1, OpenType, TrueType) und es kann bereits in der .blend existierende Objekte als Buchstaben benutzen.

Texte in Blender erlauben die Erzeugung von 2D oder 3D Texten mit allen möglichen Einstellungen. Standardmäßig sind Buchstaben flache 2 Strukturen aber sie können in Meshobjekte umgewandelt werden.

Das Bild zeigt einige Beispiele für Textobjekte in Blender.

Hinweis

Es gibt ein theoretisches Limit von 50000 Zeichen in Textobjekten, aber das Objekt wird sich langsamer verändern lassen je mehr Zeichen ein Textobjekt hat.

Textbearbeitung

Mode: Bearbeitungsmodus

Hotkey: siehe unten



Text im Bearbeitungsmodus

Bearbeitung von Text in Blender funktioniert größtenteils wie ein Texteditor. Es gibt Tastenkombinationen für Formatierungen und andere Funktionen und einige Einstellungen in den Objekteigenschaften, allerdings ist Textbearbeitung in Blender nicht so vielfältig wie in einem Texteditor:

- verlassen des Bearbeitungsmodus↵ Tab
 - macht keinen Tab sondern wechselt zwischen Objekt und Bearbeitungsmodus.
- KopierenStrgC oder der Kopieren Button
 - kopieren den markierten Text in die Zwischenablage.
- Ausschneiden StrgX oder der Ausschneiden Button
 - schneidet den markierten Text aus und speichert ihn in der Zwischenablage
- Einfügen StrgV oder der Einfügen Button
 - fügen den Text aus der Zwischenablage ein

Text löschen **Ctrl←** Backspace
um allen Text im Objekt zu löschen

Anfang und Ende **↶** Home and **↷** End
bewegt den Cursor.

Nächstes/Vorheriges Wort **Ctrl←** oder **Ctrl→**
bewegt den Cursor.

Der Zwischenspeicher funktioniert nur innerhalb Blender.

Text einfügen

Text kann auf drei Arten eingefügt werden: schreiben, aus der internen Zwischenablage oder aus einer Textdatei.

Um den text aus einer Datei zu laden muss das Text » Datei einfügen Werkzeug benutzt werden welches den Dateibrowser öffnet.

Sonderzeichen

Mode: Bearbeitungsmodus

Menu: Text » Sonderzeichen

Es gibt einige verfügbare sonderzeichen die über die `{{Shortcut|alt}}` Taste oder das Textmenü verfügbar sind.

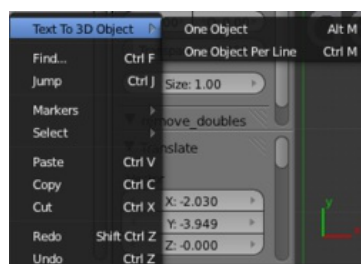
Diese sind:

AltC:	Copyright (©)	AltR:	Registered trademark (®)
AltG:	Grad (°)	AltX:	Multiplikationszeichen (×)
AltS:	das altedeutsche s (ß)	AltF:	Währungssymbol (₣)
AltL:	Britisches Pfund (£)	AltY:	Japanischer Yen (¥)
Alt1:	Hoch 1 (¹)	Alt2:	Hoch 2 (²)
Alt3:	Hoch 3 (³)	Alt:	Kreis
Alt?:	Spanisches Fragezeichen (¿)	Alt!:	Spanisches Ausrufezeichen (¡)
Alt<:	Liker Doppelpfeil («)	Alt>:	Rechter Doppelpfeil (»)

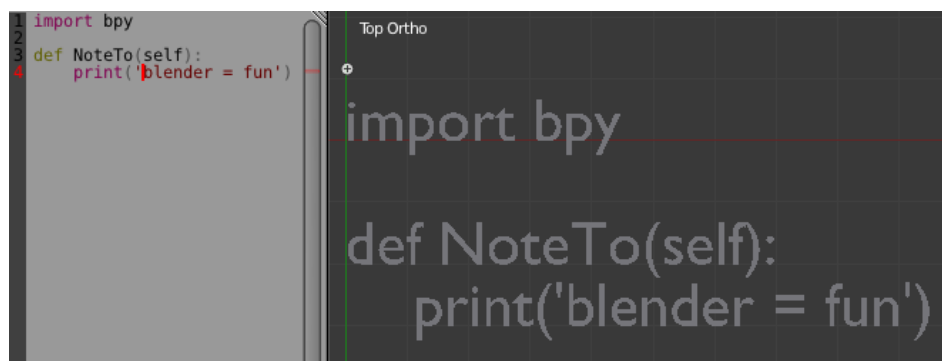
Alle Buchstaben auf der Tastatur sollten funktionieren. Falls besondere Buchstaben benötigt werden können diese als Kombination von zwei Buchstaben erhalten werden. Es muss der Grundbuchstabe und ein modifizierender zweiter Buchstabe gedrückt werden. Hier sind einige Beispiele:

A, Alt← Backspace, ~:	ä A, Alt← Backspace, ':	á A, Alt← Backspace, `:	à
A, Alt← Backspace, O:	å E, Alt← Backspace, ":	ë O, Alt← Backspace, /:	ø

Text zu Textobjekten umwandeln



Ein leichter Weg Text in Blender einzufügen ist ihn im Texteditor zu schreiben und dann in ein Textobjekt umzuwandeln.



links normaler Text, rechts in einem Textobjekt

Textauswahl



Text im Bearbeitungsmodus

Im Bearbeitungsmodus hat der Text einen Cursor der bestimmt wo neue Buchstaben eingesetzt werden. Um diesen Cursor zu bewegen können die Pfeiltasten (→/↓/←/↑) oder Page up/Page down und `↵` Home/Ende benutzt werden In Edit mode, your text has a white cursor, and as in any text editor, it determines where new chars will be inserted! You move this cursor with the arrow keys (→/↓/←/↑) or Page up/Page down und `↵` Home/→ End keys.

Das halten von `⇧` Shift beim bewegen wählt den Text aus.

Textformatierung












Schriftarten

Mode: Bearbeitungsmodus

Panel: Schriftart (Bearbeitungskontext)

Das Schriftarten Panel hat mehrere Einstellungen zum einstellen von Zeichen.

Laden und verändern von Schriftarten

 18thCtRinit.ttf	29 KB
 18thCtKurStart.ttf	89 KB
 18thCtKurTxt.ttf	84 KB
 1610_Cancellaresca_lim.TTF	116 KB
 BIRTH OF A HERO.ttf	116 KB
 CRBLATRIAL.otf	45 KB
 CRBLATRIAL.ttf	54 KB
 CRIALTRIAL.otf	59 KB
 CRIALTRIAL.ttf	76 KB
 kabog.ttf	23 KB
 kingthingspetrock.ttf	14 KB

Laden einer Typ1 Schriftart

Blender hat eine eingebaute Schriftart ("Bfont") die standardmäßig verwendet und angezeigt wird.

Um eine andere Schriftart zu laden muss der **Laden** gedrückt werden und eine gültige Schriftart ausgewählt werden.

Unix Hinweis

Schriftarten sind typischerweise unter `/usr/lib/fonts`, oder einer Variante wie `/usr/lib/X11/fonts` gespeichert, aber nicht immer. Sie können auch in anderen Orten wie `/usr/share/local` oder `/usr/local/share` und untergeordneten Ordnern sein.

Falls eine nicht gültige Schriftart ausgewählt wird, wird "Not a valid font" angezeigt werden.

Dieselbe Schriftart wird auf alle Buchstaben mit dem gleichen Stil angewendet aber eine andere ist nötig in einem anderen Schriftstil. Es werden also unterschiedliche Schriften für **Normal Kursiv**, **Fett** und **Fett und Kursiv** benötigt.

Blender legt keinen Wert darauf für welchen Stil welche Schriftart geladen wird. Es kann die gleiche Schriftart für alle modi oder komplett unabhängige Schriftarten sein.

Größe und Verschiebung

Größe

Kontrolliert die Größe des ganzen Texts (es gibt keine Möglichkeit einzelne Buchstaben zu vergrößern). Es kann aber sein das unterschiedliche Schriftarten unterschiedliche Größen haben.



Verschieben: Blender hat einen verschiebenwert von 1 , 2.59 hat einen Wert von 0

Verschieben

Kontrolliert die Schrägstellung des Texts, auch wenn das ähnlich wie Kursiv aussehen kann ist es nicht das selbe.

Objekte als Buchstaben oder Schrift

Es kann eine eigene Schrift in Blender erstellt und benutzt werden.

Das geht folgendermaßen:

- Zu allererst müssen die Buchstaben erstellt werden. Buchstaben können Objekte jedes Typs sein. Sie müssen alle diesem Namenschema folgen: gemeinsamer Anfang gefolgt von Buchstabenname (also "ft.a", "ft.b", etc.).
- Dann muss für das Textobjekt [Vertex Duplikation](#) eingeschaltet werden.
- Zurück in den Objekt Eigenschaften im Font Panel muss das Ob Familie mit dem gemeinsame Anfang ausgefüllt werden.

Jedes Mal wenn der der Buchstabe benutzt wird für den es ein Objekt mit dem angegebenen Anfang gibt, wird das Objekt in der linken unteren Ecke des Buchstabens platziert. Alle Ursprungsobjekte werden weiterhin dargestellt aber nicht mehr gerendert.

Text entlang einer Kurve

Mit dem [Kurvenmodifikator](#) kann Text einer Kurve folgen.



Text der einer Kurve folgt.

Um Text einer Kurve folgen zu lassen muss es sich dabei um ein Mesh handeln. Objekte werden mit AltC in Meshobjekte umgewandelt.

Hinweis

Es gibt auch ein anderes Feature um Text der einer Kurve folgt zu erzeugen aber der Kurvenmodifikator hat einen besseren Funktionsumfang.

Unterstreichen

Unterstreichen

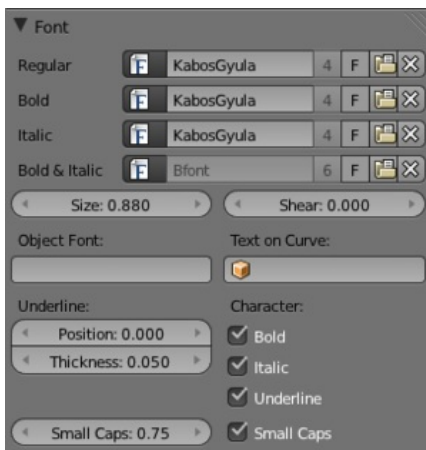
wird mit dem entsprechenden Knopf ein- oder ausgeschaltet.

Position

erlaubt der vertikale verschieben der Linie.

Dicke

stellt die Dicke der Linie ein.



Einstellungsbuttons

Buchstabenstile



Fett

Fett

Kursiv

Die Stilwechsel in Blender funktionieren nicht wie Stilwechsel in anderen Texteditoren, wo eine existierende Schrift dann im neuen Stil dargestellt wird. In Blender bewirkt das drücken eines Stilwechselknopfs das Benutzen einer neuen Schriftart, die den gewünschten Stil haben *kann* aber nicht muss. Siehe [#Schriftarten](#) für Informationen wie man eine neue Schriftart (und damit Schriftstil) in Blender lädt.

Kapitälchen

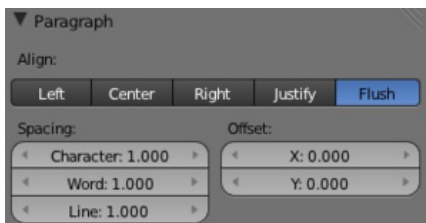
Kapitälchen funktionieren wie erwartet, in der ursprünglichen Schrift. Die Größen der kleinen Kapitälchen kann mit dem **Kapitälchenskalierung** verändert werden.

Groß- und Kleinschreibung

Die Groß- und Kleinschreibung kann durch auswählen des Textes und drücken der Buttons für **Zu Großs.** und **Zu Kleins.**.

Absatz

Das Absatz Panel enthält Einstellungen über die Ausrichtung und die Abstände im Text.



Das Absatz Panel

Orientierung

Linksbündig

Text wird linksbündig vom Objektzentrum weg geschrieben.

Zentriert

Der Text wird auf den Objektsprung zentriert geschrieben.

Rechtsbündig

Text wird rechtsbündig vom Objektzentrum weg geschrieben.

In Textrahmen**Justify**

Fängt nur eine neue Linie an wenn die alte von einer Wordtrennung beendet wird (also nicht ↵ Enter).

Flush

Macht immer einen Zeilenumbruch auch während der Eingabe.

Abstände**Buchstabenabstand**

Hiermit kann der Abstand zwischen Buchstaben skaliert werden.

Wort

Ein Faktor mit dem Leerzeichen skaliert werden.

Zeile

Ein Faktor mit dem der Abstand zwischen Zeilen skaliert wird.

Abstand**X Abstand und Y Abstand**

Positioniert den Text mit einem X und Y Abstand zum Objektzentrum als normale Einstellung. Bei Textrahmen gilt dies für den ganzen Textrahmen.

Form

Mode: Objekt- oder Bearbeitungsmodus

Panel: Kurve oder Oberfläche

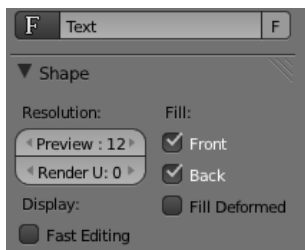
Wie man im Kurven und Oberflächen Panel sehen kann, haben Texte die gleichen Optionen wie Kurven.

Auflösung**Vorschau**

Siehe die [Auflösung](#) in der 3D Ansicht.

Render

Siehe die [Auflösung](#) im Render.



Die Formeinstellungen

Schnelles Bearbeiten

Schaltet das füllen von Kurven im Bearbeitungsmodus aus.

Füllung

Die Füllung kontrolliert wie Textkurven sich verhalten wenn sie extrudiert werden oder gebevelt werden.

Vor

Füllt die Vorderseite der Oberfläche

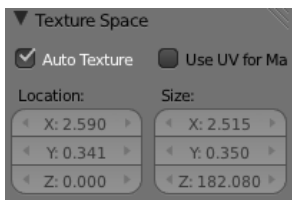
Rück

Füllt die Rückseite der Oberfläche

Fülle verformte

Füllt die Kurven nach Anwendung der Form keys und Modifizierer.
Fills the curves after applying shape keys and modifiers.

Texturen



Textur Einstellungen

Benutzt UV zum Positionieren

Benutzt UV Koordinaten für die generierte Position.

Automatik

Führt das Positionieren der Textur automatisch aus.

Geometrie

Textobjekte haben die gleichen Eigenschaften wie [Kurven](#).

Meta Objekte

Mode: Objekt oder Bearbeitungsmodus

Hotkey: ⇧ ShiftA

Menu: Hinzufügen » Meta

Meta Objekte bilden "implizite Oberflächen", was so verstanden werden sollte das sie nicht "explizit" von Vertices oder Kontrollpunkten definiert werden, so wie Meshes oder Oberflächen. Sie werden dynamisch aus Formeln erzeugt.

Ein sehr gutes Erkennungsmerkmal von Metaobjekten ist das die flüssig oder tonern wirken und eine abgerundete Form haben. Wenn zwei Metaobjekte sich nah genug kommen interagieren und verbinden sie sich wie zum Beispiel Wassertropfen in Schwerelosigkeit.

Jedes von ihnen wird von zugrundeliegenden Formeln erzeugt.

Typischerweise werden Metaobjekte für Spezialeffekte oder als Modelliergrundlage benutzt. So kann die Grundform eines Körpers aus Metaobjekten erzeugt und dann in ein Mesh umgewandelt werden. Außerdem sind sie sehr effizient für Ray-tracing.

Allerdings verhalten sie sich im Objektmodus etwas anders als gewöhnliche Objekte.

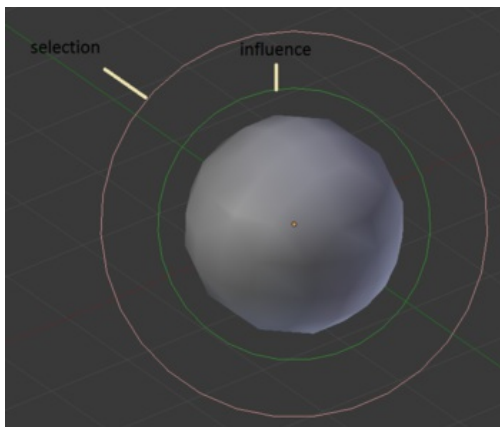
Grundformen

Es gibt fünf vordefinierte Metaformen:

- Meta Kugel erzeugt ein Meta mit einem Punkt als zugrundeliegende Struktur.
- Meta Röhre erzeugt ein Meta mit einer Linie als zugrundeliegende Struktur.
- Meta Ebene erzeugt ein Meta mit einer Ebene als zugrundeliegende Struktur.
- Meta Ellipsoid erzeugt ein Meta mit einem Ellipsoid als zugrundeliegende Struktur.
- Meta Quader erzeugt ein Meta mit einem Quader als zugrundeliegende Struktur.

Visualisierung

Im Objektmodus wird das berechnete Mesh angezeigt, zusammen mit einem Auswahlkreis.

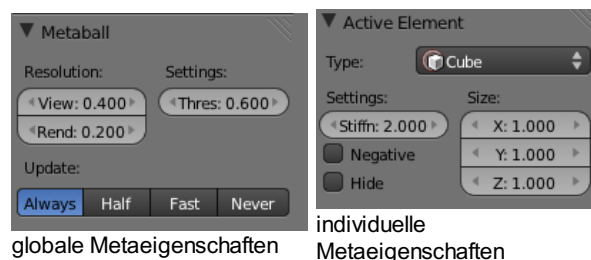


Meta Kugel

Im Bearbeitungsmodus wie er im Bild zu sehen ist, wird ein Metaobjekt als Mesh mit zwei farbigen Kreisen dargestellt, einer für die Auswahl und ein zweiter für direkte Kontrolle der Steifigkeit des Metaobjekts.

Meta Kugel Optionen

Alle Metaobjekt in einer Szene interagieren mit einander. Die Einstellungen in der Metakugelsektion gelten für alle Meta Objekte. Im Bearbeitungsmodus erscheint ein Panel für das aktive Element zum Bearbeiten einzelner Metaelemente.



globale Metaeigenschaften

individuelle
Metaeigenschaften

Auflösung

Die Auflösung kontrolliert die Vertexdichte des Objektes das vom Metaobjekt erzeugt wird.

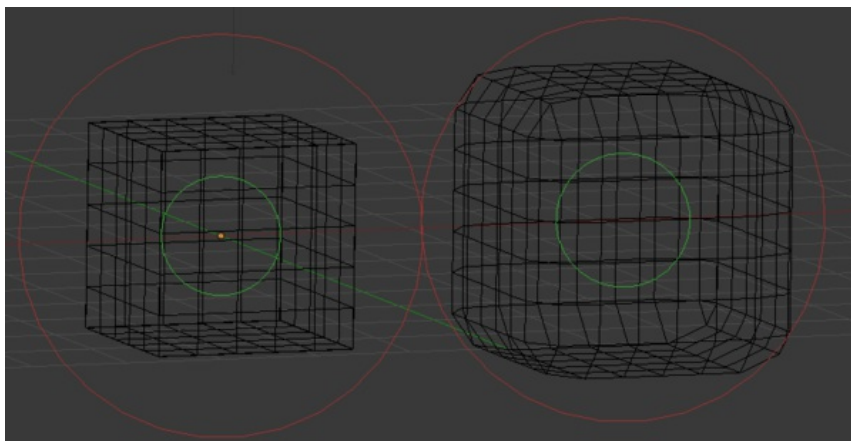
Ansicht

Die Vertexdichte in der 3D Ansicht(von 0.05 bis 1.0).

Render

Die Vertexdichte des gerenderten Objekts(von 0.05 bis 1.0).

Eine Möglichkeit die einem Metaobjekt zugrundeliegende Struktur sichtbar zu machen ist die Auflösung herunter zu stellen oder den Grenzwert hoch zu setzen und die Steifigkeit etwas über den Grenzwert zu setzen. Der unten gezeigte Metaquader hat folgende Konfiguration: Auflösung: 0.410, Grenzwert: 5.0, Steifigkeit: 5.01.



Links: die zugrundeliegende Struktur, Rechts: die Form

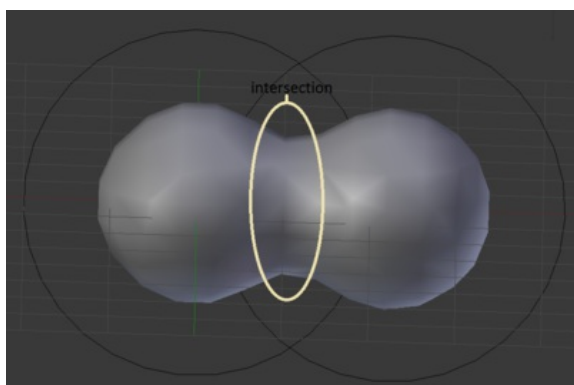
Grenzwerteinfluss

Mode: Objekt oder Bearbeitungsmodus

Panel: Metakugel (Bearbeitungskontext, F9)

Der Grenzwert Grenzwert definiert wie die Oberfläche eines Metaobjekts andere Metaobjekte beeinflusst. Es kontrolliert das "Feldlevel" auf dem die Oberfläche erzeugt wird. Die Einstellung ist global für eine Gruppe von Metaobjekten. Wenn der Grenzwert erhöht wird, steigt der Einfluss von Metas aufeinander.

Es gibt positiven und negativen Einfluss, der im Panel des aktiven Elements im Bearbeitungsmodus eingestellt werden kann. Positiver Einfluss wirkt wie Anziehung und negativer wie abstoßender Einfluss auf die Meshes. Ein negative Meta drückt positive Meshes von Metaobjekten von sich weg.



Positiver Einfluss

Positiv eingestellte Meshes von Metaobjekten ziehen sich gegenseitig an, wenn sich ihre Einflusskreise schneiden.

Aktualisierung

Beim transformieren von Metaobjekten gibt es vier Darstellungsmodi in den Aktualisierungs Knöpfen des Metakugel Panels:

- Immer - stellt das Meta während der Transformation vollständig dar.
- Halbe Auflösung - während Transformationen werden Metas nur mit halber Auflösung dargestellt.

- Schnell - Metas werden bei Transformationen nicht dargestellt.
- Nie - Metas werden nicht dargestellt und nur gerendert.

Meta Strukturen

Technische Details

Eine mehr technische Definition von Metaobjekten kann als "richtungsgebende Struktur" gegeben werden die als Quelle eines statischen Feldes angesehen werden kann. Das Feld kann positiv oder negativ sein und dementsprechend wird das erzeugte Feld von benachbarten Feldern angezogen oder abgestoßen werden.

Die implizierte Oberfläche ist definiert als die Oberfläche an der das 3D Feld einen bestimmten Wert annimmt. Im Beispiel einer Metakugel, dessen richtungsgebende Struktur ein Punkt ist, wird ein isotropisches (in allen Richtungen identisches) Feld erzeugt und die Oberflächen sind Kugeln mit dem richtungsgebenden Punkt als Mittelpunkt.

Metaobjekte sind nichts anderes als mathematischen Formeln die eine logische Operation aufeinander ausführen und die addiert oder subtrahiert werden können. Diese Methode wird auch "Konstruktive Solid Geometry" (CSG/KSG) genannt. Aufgrund seiner mathematischen Natur braucht KSG sehr wenig Speicher, aber viel Rechenleistung.

Grundlegende Struktur

Mode: Bearbeitungsmodus

Panel: Metakugel Werkzeuge (Bearbeitungskontext F9), Transformationseigenschaften

Blender hat fünf Typen von Metaobjekten, die im Bearbeitungsmodus in den Metakugel Werkzeugen oder in den Transformationseinstellungen verändert oder eingestellt werden können.

Kugel (Punkt, 0-dimensionale Struktur)

Das einfachste Meta, ohne zusätzliche Einstellungen, es erzeugt eine isotropische Oberfläche.

Tube (Gerade, 1-dimensionale Struktur)

Dieses Meta wird von einer geraden einer bestimmten Länge erzeugt und ergibt eine zylindrische Oberfläche mit abgerundeten Enden.

- dx: Die Länge der Linie (und damit der Röhre), standardmäßig auf 1 gesetzt..

Ebene (rechteckige Fläche, 2-dimensionale Struktur)

Dieses Meta wird von einer Ebene erzeugt und ergibt eine Ebene mit einer Dicke und abgerundeten Ecken und Kanten.

- dx: Die x-Ausdehnung des Rechtecks.
- dy: Die y-Ausdehnung des Rechtecks.

Elipsoid (Elipsoides Volumen, 3-dimensionale Struktur)

Dieses meta wird von einem elliptischen Körper verursacht und erzeugt dementsprechend ebenfalls einen elliptischen Körper.

- dx: Die x-Ausdehnung des Rechtecks.
- dy: Die y-Ausdehnung des Rechtecks.
- dz: Die z-Ausdehnung des Rechtecks.

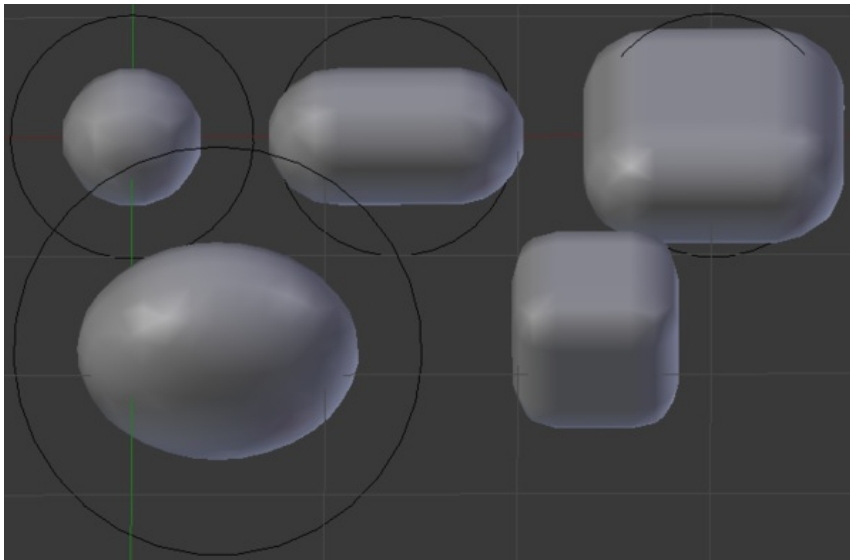
Standardmäßig sind alle Längen gleich lang und erzeugen eine Kugel.

Quader (3-dimensionale Struktur)

Dieses Meta wird von einem Quader erzeugt und hat dementsprechend die Form eines Quaders mit abgerundeten Ecken und Kanten.

- dx: Die x-Ausdehnung des Rechtecks.
- dy: Die y-Ausdehnung des Rechtecks.
- dz: Die z-Ausdehnung des Rechtecks.

Standardmäßig sind alle Längen gleich lang und erzeugen einen Würfel.



Die fünf Metagrundkörper

Leere Objekte

Ein "leeres" Objekt ist ein Null Objekt, es enthält keine wichtigen Eigenschaften außer seiner Position. Leere Objekte können für viele Dinge benutzt werden für die andere Objekte unnötig detailliert sind.

Einstellungen

einfache Achsen

das leere Objekt wird als Koordinatenkreuz dargestellt.

Pfeile

das leere Objekt wird als Koordinatenkreuz in X,Y und Z Richtung mit Beschriftung und einem Pfeil an den Achsenenden dargestellt.

einzelner Pfeil

das leere Objekt wird als einzelner Pfeil in +Z Richtung dargestellt.

Kreis

das leere Objekt wird als Kreis in der XZ Ebene dargestellt.

Würfel

das leere Objekt wird als Würfelgitter dargestellt.

Kugel

das leere Objekt wird mit 3 Kreisen, am Äquator und auf zwei Längengeraden, dargestellt sodass ein Kugeleindruck entsteht.

Kegel

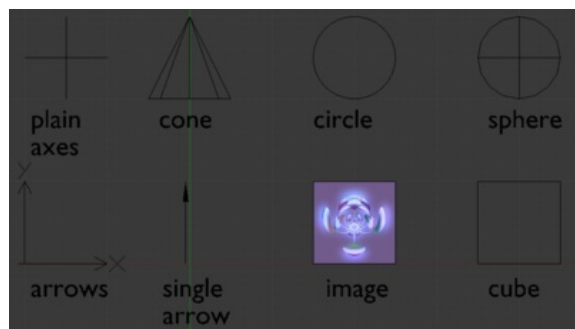
das leere Objekt wird als Kegel, dessen Spitze in +Y Richtung zeigt, dargestellt.

Bild

das leere Objekt wird als ein beliebiges Bild dargestellt. Dies kann als Referenzbildquelle benutzt werden und ist eine Alternative zu [Hintergrundbildern](#). Das leere Objekt ist im Gegensatz zu Hintergrundbildern immer sichtbar.

Größe

stellt die lokale Größe ein mit der das leere Objekt dargestellt wird. Es verändert nicht die Skalierung sondern nur die Darstellung.



Die verschiedenen Darstellungstypen

Benutzung und Funktion

Leere Objekte dienen als Transformationsreferenz die nicht gerendert wird. Daher sind leere Objekte sehr nützliche und wichtige Objekte. Hier sind ein Paar Anwendungsbeispiele:

Übergeordnetes Objekt für eine ObjektgruppeParent object for a group of objects

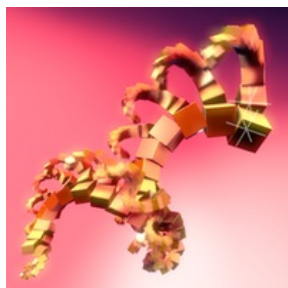
- Ein leeres Objekt kann anderen Objekten unter- oder übergeordnet werden und diese so Zusammenfassen.

Beschränkungsreferenz

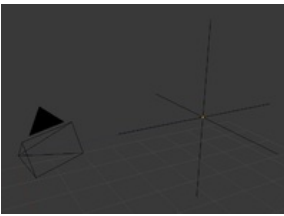
- Ein leeres Objekt kann auch eine Beschränkung für Normal- oder Knochenbeschränkungen sein, was dem Benutzer viel mehr Kontrolle über die Beschränkung gibt. Ohne die Einstellungen einzugeben kann einfach das leere Objekt an die gewünschte Position verschoben werden.

Arrayabstand

- Ein leeres Objekt kann dazu benutzt werden den Abstand zu markieren auf dem ein Arraymodifizierer das Array erzeugt. Einstellungen werden dann durch verschieben des leeren Objekts vorgenommen.



Ein leeres Objekt steuert ein Array



Ein leeres Objekt steuert
eine Beschränkung

Andere Zwecke

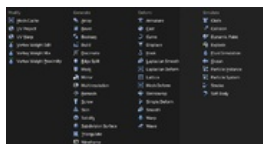
- Platzhalter
- Rigging Steuerung
- Depth of Field - Kameraentfernungen

Gruppen

Gruppen werden [hier](#) vollständig erklärt.

Modifikatoren

Panel: Modifikatoren



Modifikatoren Menü

Objektmodifikatoren sind automatische Operationen, die ein Objekt auf eine nichtverändernde Weise beeinflussen. Mit Modifikatoren können viele Effekte automatisch ausgeführt werden, die manuell sehr mühsam durchzuführen wären (zum Beispiel Oberflächen zu unterteilen) und ohne die Basistopologie des Objekts zu verändern. Modifikatoren funktionieren durch die Änderung der Anzeige und des Renderns eines Objekts, aber nicht durch das Ändern der eigentlichen Objektgeometrie. Verschiedene Modifikatoren können zu einem Objekt hinzugefügt werden um einen [Modifikatorstapel](#) zu bilden, und ein Modifikator kann man anwenden, wenn man möchte, dass seine Änderungen dauerhaft übernommen werden sollen.

Es existieren vier Typen von Modifikatoren:

Modifizieren

Die Modifizieren-Gruppe der Modifikatoren sind Werkzeuge die ähnlich zu denen von Transformatoren sind, die sich aber nicht unmittelbar auf die Form des Objektes auswirken, sondern eher wie andere Daten wie Vertexgruppen sind...

[UV-Projektion](#)

Projiziert UV-Koordinaten auf einen Mesh.

[Vertex Weight](#)

Bearbeiten einer Vertexgruppe eines Meshes auf verschiedene Weisen.

Generieren

Die Generieren-Gruppe der Modifikatoren sind konstruktive Werkzeuge die entweder das allgemeine Erscheinungsbild ändern oder dem Objekt automatisch neue Geometrie hinzufügen.

[Array \(Gruppe\)](#)

Anlegen einer Gruppe außerhalb eines Basis-Meshes und ähnliche sich wiederholende Formen.

[Bevel \(Fasen\)](#)

Anlegen einer Fase auf ein ausgewähltes Mesh-Objekt.

[Boolean](#)

Verbinden/entfernen/schneiden des Meshes mit einem Anderen.

[Build \(Bauen\)](#)

Zusammenstellung Stück für Stück eines Meshs während der Animation.

[Decimate \(Dezimieren\)](#)

Reduzieren der Polygonanzahl des Meshes.

[Edge Split \(Kanten teilen\)](#)

Hinzufügen scharfer Kanten zum Mesh.

[Mask \(Maske\)](#)

Verbergen einiger Teile des Meshs.

[Mirror \(Spiegel\)](#)

Spiegeln eines Objektes ein oder mehrere Male um seine Achse, sodass das resultierende Mesh symmetrisch ist, und man nur noch die Hälfte, ein Viertel, oder ein Achtel des Meshes modellieren muss.

[Multiresolution \(Mehrfach Auflösung\)](#)

Formen des Meshs bei verschiedenen Stufen der Auflösung.

[Screw \(Schraube/Helix\)](#)

Genieren von Geometrie im Schrauben-Muster aus einem einfachen Profil. Ähnlich dem Screw Werkzeug im Mesh-Bearbeitungskontext.

[Solidify \(Verdicken\)](#)

Zum Verleihen von Tiefe bei Meshes.

[Subdivision Surface \(Oberflächen unterteilen\)](#)

Glätten der Oberfläche durch Erstellen von interpolierter Geometrie.

[Skin \(Haut\)](#)

Automatisch generierte Topologie.

Verformen

Die Deformieren-Gruppe der Modifikatoren ändert nur die Gestalt eines Objekts, und ist für Meshes und oft auch Texte, Kurven, Oberflächen und Gitter verfügbar.

[Armature \(Skelett\)](#)

Verwendet Knochen um ein Objekt zu verformen und zu animieren.

[Cast](#)

Wechselt die Gestalt eines Meshes, einer Oberfläche oder eines Gitters mit einem Bereich, Zylinder oder Würfel aus.

[Curve \(Kurve\)](#)

Biegt einen Gegenstand entlang einer Kurve als Führung.

[Displace \(Versetzen\)](#)

Verformt ein Objekt durch Verwendung einer Textur.

[Hook \(Haken\)](#)

Fügt einen Haken zu den Vertice(s) oder Anfassern hinzu, um diese von außerhalb zu ändern.

[Lattice \(Gitter\)](#)

Verwenden eines Gitter-Objekts, um das Objekt zu verformen.

[Mesh Deform \(Mesh deformieren\)](#)

Erlaubt es, das Objekt durch ändern der Form eines anderen Meshes zu verformen, durch Verwendung eines "Mesh Deform Cage" (wie bei der Verwendung eines Gitters).

[Shrinkwrap \(Schrumpfen/Umhüllen\)](#)

Erlaubt es, das Objekt zu verkleinern/zu umhüllen bis zur Oberfläche des Ziel-Mesh-Objekts.

[Simple Deform \(Einfaches Verformen\)](#)

Wendet einige erweiterte Verformungen auf das Objekt an.

[Smooth \(Weich\)](#)

Glättet die Geometrie eines Meshes ähnlich dem Smooth-Werkzeug im Mesh-Bearbeitungskontext.

[Warp \(Wölben\)](#)

Wölbt ein Mesh, durch Festlegen von zwei Punkten, zwischen denen sich das Mesh befindet.

[Wave \(Welle\)](#)

Verformt ein Objekt zu einer animierten Wellenform.

Simulieren

Die Simulations-Gruppe von Modifikatoren steuert Simulationen. In den meisten Fällen werden diese Modifikatoren automatisch zum Modifikatorstapel hinzugefügt, wenn ein Partikelsystem oder eine Physik-Simulation eingeschaltet wird und ihre einzige Aufgabe ist es, den Platz im Modifikator-Stack festzulegen, der als Basisdaten vom Werkzeug genutzt wird, dass sie vertreten. Allgemein sind die Attribute dieser Modifikatoren in gertennten Fenstern erreichbar.

[Cloth \(Kleidung\)](#)

Simuliert die Eigenschaften eines Kleidungsstücks. Es wird in den Modifikatorstapel eingefügt, wenn ein Mesh als Kleidung gekennzeichnet wird.

[Collision \(Kollision\)](#)

Simuliert den Zusammenstoß zweier Objekte.

[Explode \(Explodieren\)](#)

Das Mesh explodiert mit Hilfe eines Partikelsystems.

[Fluid \(Flüssigkeiten\)](#)

Das Objekt ist Teil einer Flüssigkeits-Simulation. Der Modifikator wird hinzugefügt, wenn ein Mesh als Flüssigkeit gekennzeichnet wird.

[Particle Instance \(Partikel Instanzen\)](#)

Bringt ein Objekt dazu sich wie ein Partikel zu verhalten, benutzt dazu aber die Mesh-Form.

[Particle System \(Partikel Systeme\)](#)

Vertritt ein Partikelsystem im Stapel, das verwendet wird, wenn ein Partikelsystem zum Objekt hinzugefügt wird.

[Smoke \(Rauch\)](#)

Simuliert realistischen Rauch.

[Soft Body \(Weiche Körper\)](#)

Das Objekt ist weich und elastisch. Der Modifikator wird hinzugefügt, wenn das Objekt als Weichkörper gekennzeichnet wird.

[Dynamic Paint \(Dynamische Farbe\)](#)

Lässt ein Objekt oder ein Partikelsystem ein Material auf ein anderes Objekt malen.

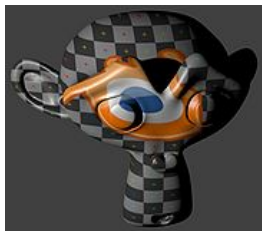
[Ocean \(Ozean\)](#)

Erzeuge schnell einen realistischen animierten Ozean.

UV-Projektion Modifikator

Modus: Any mode

Panel: Modifiers (Generate)

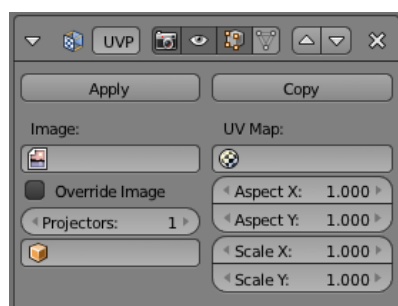


Projektion des Blender-Logos auf Suzanne.

UV-Projektion arbeitet wie ein Diaprojektor. Es projiziert ein UV-Bild von der -Z-Achse aus auf bis zu 10 Objekte und wendet es auf die Objekte an so wie das "Licht" sie trifft. Wahlweise kann es die [Flächentextur](#) des Objekts überschreiben.

[Herunterladen eines Beispiels](#)

Options



UV layer (UV-Ebene)

Welche [UV-Ebene](#) geändert werden soll. Default ist die aktive Renderebene.

Image (Bild)

Das [Bild](#), das mit dem Modifikator verbunden ist. Nicht notwendig: Sie können einfach eine UV zur Benutzung an anderer Stelle projizieren. *Überschreibbild* unten beschreibt wie das Bild benutzt wird.

Override Image (Überschreibbild)

- Wenn "true", ersetzt [Flächentexturen](#) auf allen Vertexen mit dem Bild. Dies bewirkt, dass das Bild wiederholt wird, was normalerweise unerwünscht ist.
- Wenn "false" ist der Modifizikator begrenzt auf Flächen mit dem Bild als Flächenbild.

Projectors (Projektoren)

<Objects> (Objekte)

Bis zu 10 Projektorobjekte werden unterstützt. Jede Fläche wählt den Projektor, der am nächsten seines [Flächennormals](#) ist. Projektoren strahlen von der -Z-Achse (z.B. direkt nach unten von einer Kamera oder Lampe aus).

Wenn der Projektor eine Kamera ist, dann folgt die Projektion ihrer perspektivischen/senkrechten Einstellung.

Aspect X/Y (Seitenverhältnis)

Scale X/Y (Größe)

Diese erlauben einfache Änderungen des Bildes.


Benutzung

UV-Projektion ist sehr gut geeignet um Scheinwerfer aller Art zu erstellen und auch um [W:Aufkleber](#) zu erstellen und damit Wiederholungen zu unterbrechen.

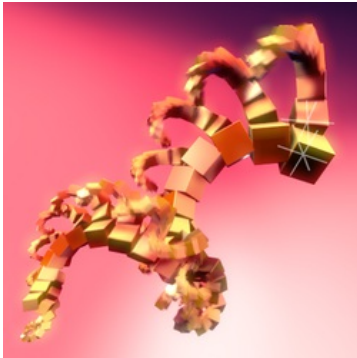
Das Image-Attribut wird normalerweise nicht benutzt. Stattdessen wird eine [Textur](#) zum [Material](#) des Objekts hinzugefügt, die auf die vom Modifikator definierte [UV-Ebene](#) projiziert wird. Dies verhindert die Wiederholung der Textur durch das Setzen von *Texture > Image Mapping > Extension auf Clip*.

Array Modifikator

Modus: Objektmodus

Panel: Eigenschaften Fenster -> Kontext Knopf Modifikatoren 

Beschreibung



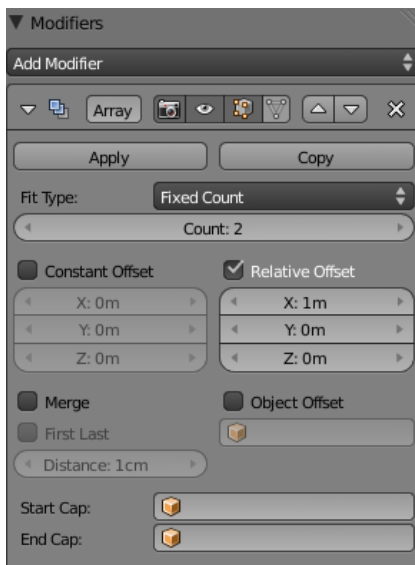
Multidimensionales Array mit animierter Bewegungsunschärfe.

Der Array Modifikator erstellt ein Array (deutsch: Datenfeld) von Kopien des Basisobjekts, mit einer verschobenen Position für jede Kopie in einer Anzahl von möglichen Wegen. Vertices in benachbarten Kopien können auf der Grundlage von einer Distanz zur Verschmelzung verschmolzen werden, was es erlaubt Frameworks für weiche Unterteilungen zu generieren.

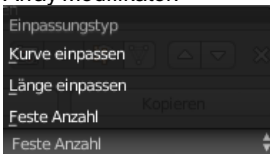
Dieser Modifikator kann nützlich sein für die Kombination von kacheligen Meshes für die schnelle Entwicklung von großen Szenen. Auch für die Erstellung von sich komplex wiederholenden Formen.

Mehrere Array Modifikatoren für ein Objekt können zur selben Zeit aktiv sein, zum Beispiel zum Erstellen von komplexen 3 dimensional Konstrukten.

Optionen



Array Modifikator.



Einpassungstyp Menü.

Einpassungstyp Menü

Kontrolliert wie die Länge des Arrays festgestellt wird. Es existieren drei Möglichkeiten, die Aktivierung der Einstellungen für Kurve, Länge oder der Anzahl:

- Kurve einpassen – Generiert genügend Kopien um in die festgelegte Länge des Kurvenobjektes zu passen, welche in Kurve angegeben ist.
- Länge anpassen – Generiert genügend Kopien um in die festgelegte Länge zu passen.
- Fixed Count – Generiert die Anzahl an Kopien die in der Anzahl angegeben worden sind.

Kurve

Das Kurven Objekt dass für Kurve einpassen verwendet werden soll.

Länge

Die Länge die für Länge anpassen genutzt werden soll.

Anzahl

Die Anzahl an Duplikaten die für die Feste Anzahl genutzt werden soll.

Hinweise

- Kurve einpassen und Länge anpassen nutzen beide die lokale Koordinatensystem Größe des Basisobjekts, was heißt dass die Skalierung des Basisobjektes im Objektmodus nicht die Anzahl der Kopien die vom Array Modifikator generiert werden verändert.
- Länge anpassen uses the local coordinate system length of the curve, which means that scaling the curve in Objektmodus will not change the number of copies generated by the Array modifier.
- Die Anwendung von Skalierung mit Hilfe des Knopfs Skalierung anwenden kann für jede einzelne Option hilfreich sein.

Konstanter Versatz, X, Y, Z

Fügt eine konstante Komponente zur Verschiebung zu der Position des Objektes hinzu. X, Y und Z Konstanten Komponenten können angegeben werden.

Relativer Versatz, X, Y, Z

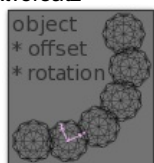


Relatives Versatz

Beispiel.

Fügt eine Adds a translation equal to the object's bounding box size along each axis, multiplied by a scaling factor, to the offset. X, Y and Z scaling factors can be specified. See (*Relative offset example*).

Objektversatz



Objektversatz

Beispiel.

Fügt eine Transformation von einem Objekt (relativ zum aktuellen Objekt) zu der Verschiebung hinzu. Siehe (*Objektversatz Beispiel*). Es ist eine gute Methode ein Leeres Objekt das zentriert oder nah an dem Startobjekt liegt zu verwenden. Durch die Rotation von diesem Leeren Objekt kann ein Kreis, oder eine Helix an Objekten erstellt werden.

Zusammenführen

If enabled, vertices in each copy will be merged with vertices in the next copy that are within the given distance.

Erste Letzte

If enabled **and** Zusammenführen aktiviert ist, vertices in the first copy will be merged with vertices in the last copy (this is useful for circular objects, see (*First Last merge example*)).



Subsurf discontinuity caused by not merging vertices between first and last copies (First Last off).

First Last merge example.



Subsurf discontinuity eliminated by merging vertices between first and last copies (First Last on).

Abstand

Kontrolliert den Abstand für das Zusammenführen.

Start cap

Das Mesh Objekt das als Start Cap verwendet werden soll. Eine einzelne Kopie dieses Objektes wird zu "Beginn" des Arrays platziert – als ob es sich an Position **-1** befinden würde, z.B. eine "Array Stufe" vor der ersten "regulären" Array Kopie. Wenn Zusammenführen aktiviert ist, und sich das Start cap nahe genug der ersten Kopie befindet, werden diese zusammengeführt.

End cap

Das Mesh Objekt das als End Cap verwendet werden soll. Eine einzelne Kopie dieses Objektes wird zu "Ende" des Arrays platziert – als ob es sich an Position **n+1** befinden würde, z.B. eine "Array Stufe" nach der ersten "regulären" Array Kopie. Und wie Start cap, kann es mit der letzten Kopie zusammengeführt werden...

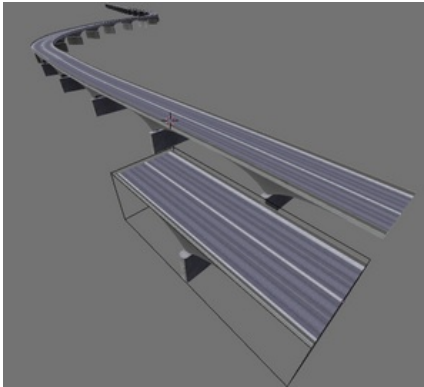
Tipps

Berechnung der Verschiebung

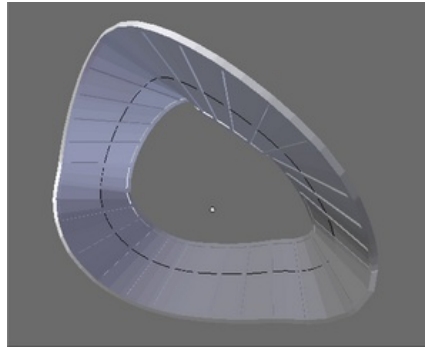
The transformation applied from one copy to the next is calculated as the sum of the three different components (Relative, Constant and Object), all of which can be enabled/disabled independently of the others. This allows, for example, a relative offset of **(1, 0, 0)** and a constant offset of **(0.1, 0, 0)**, giving an array of objects neatly spaced along the X axis with a constant **0.1BU** (Blender Units) between them, whatever being the original object's size.

Beispiele

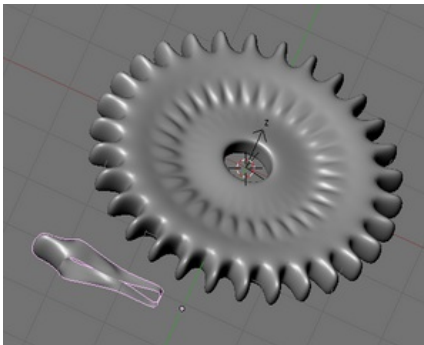
Mechanisch



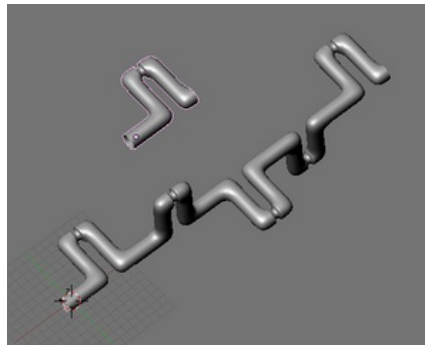
Eine Brücke, erstellt aus einem flächendeckenden Mesh.



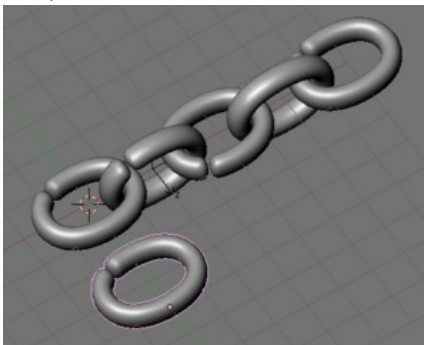
Eine Spur.
[Beispiel blend Datei](#)



Ein Zahn, erstellt aus einem einzelnen Segment.
[Beispiel Blend Datei](#)



Eine Kurbelwelle.
[Beispiel blend Datei](#)

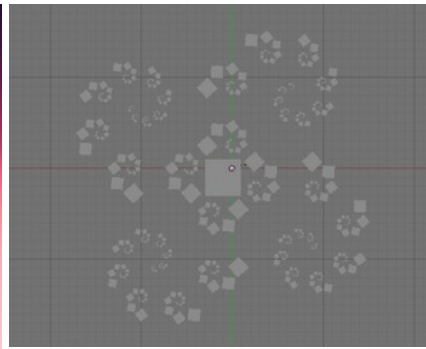


Eine Kette, erstellt aus einer einzigen Verknüpfung.
[Beispiel blend Datei](#)

Fraktal

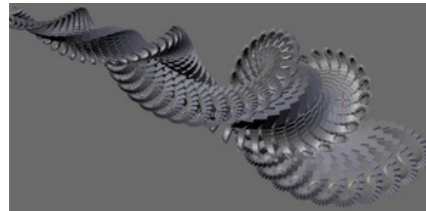


Multidimensionales Array mit animierter Bewegungsunschärfe.



Ein fraktal-ähnliches Bild, erstellt mit mehreren Array Modifikatoren die auf einen Würfel angewandt worden sind.

[Beispiel blend Datei](#)

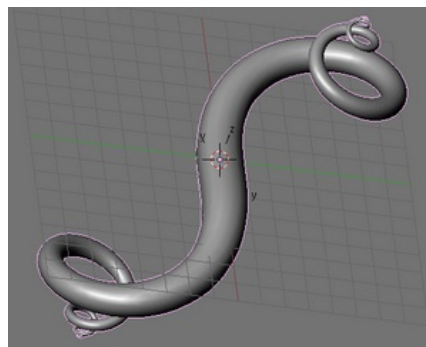


Ein fraktales Bild einer Farn, erstellt mit 2 Array Modifikatoren und 1 Spiegel der auf den Würfel angewandt worden ist.

Organisch

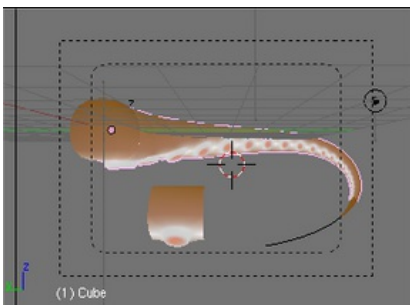


Unterteiltes Würfel Array mit 1 Objekt Verschiebung, 4 Würfeln und einer hohen Vertex Verbindungs-Einstellung um den Effekt von Skinning zu erreichen.



Eine doppelte Spirale, erstellt ausA double spiral, erstellt mit 2 Array Modifikatoren und einem Oberflächenunterteilung-Modifikator angewandt an einen Würfel. Wie oben, ist die Schwelle der Vertex Zusammenführung sehr hoch gesetzt worden um den Effekt von Skinning zu erreichen.

[Sample blend file](#)



Ein Tentakel, erstellt mit einem Array Modifikator gefolgt von einem Kurve Modifikator. Das Segment im Vordergrund ist das Basis Mesh für den Tentakel; der Tentakel wird durch zwei speziell modellierte Objekte begrenzt, die vom selben Kurve Objekt deformiert worden sind wie der Hauptteil des Tentakel.

[Sample blend file](#)

Tutorials


Tutorials

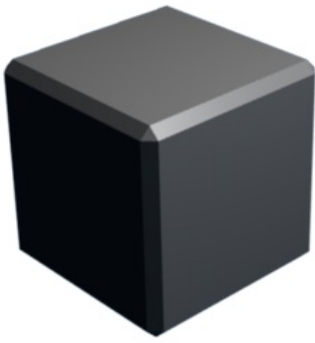
- [Neal Hirsig's Array Modifier Screencast](#)
- [Creating A Double Helix With Modifiers](#)

The 'Double Helix' tutorial explains the Array modifier. It is for an old Blender Version (2.44) but except for the keyboard shortcuts it is still valid.

Fase

Mode: Objektmodus

Panel: Eigenschaften Fenster -> Kontextknopf Modifiers 



Standard Fase

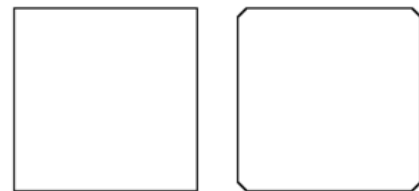
Der Fasenmodifikator erstellt Fasen am angewendeten Objekt, wobei genaue Einstellungen vorgenommen werden können.

Der Fasenmodifikator ist eine nicht destruktive Alternative zur [Fasen Operation](#).

Das Bild "normales Quadrat" zeigt ein normales Quadrat. Das Bild "Quadrat mit Fasen" zeigt ein Quadrat mit Fasen in den Ecken.

Auch wenn diese 2D Quadrate sind, zeigen sie gut was der Fasenmodifikator mit Meshes macht.

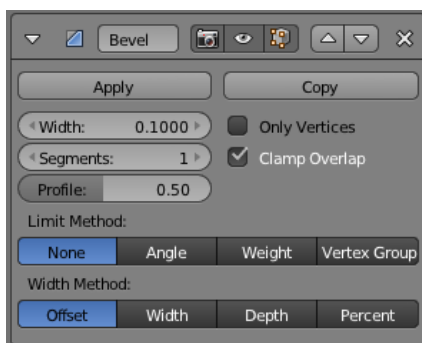
Das Bild "Standard Fase" zeigt einen Würfel mit einem Fasenmodifikator unter Standardeinstellungen.



normales Quadrat.

Quadrat mit Fasen

Optionen



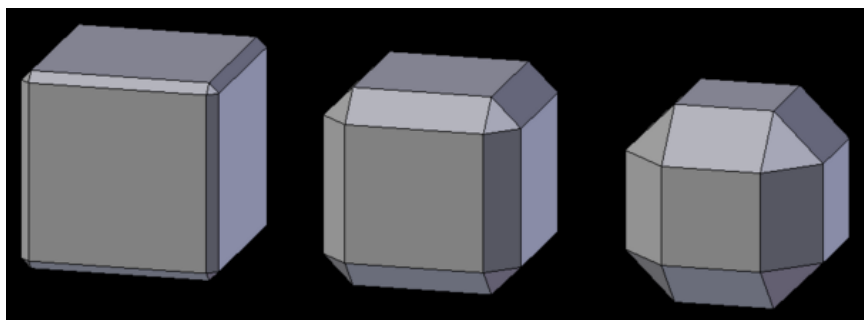
Faseneinstellungspanel.

Breite

Die Breite ist ein Skalierwert für die Fasen (von 0 bis 1) und beeinflusst wie Breit die Fasen sind die erzeugt werden.

Hinweis

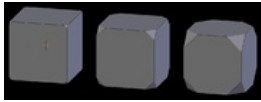
Beim benutzen von Einheiten ist dieser Wert Einheiten abhängig. Es muss also auf die richtige Größenordnung geachtet werden.



Drei Würfel mit den Fasen Breiten "0,1" , "0,3" und "0,5".

nur Vertices

Der "nur Vertices" Knopf verändert die Funktionsweise des Modifikators, sodass die Veränderung nur noch von Vertices abhängt und nicht mehr von Kanten.



Drei Würfel mit den Fasen Breiten "0,1" , "0,3" und "0,5" und "nur Vertices" eingeschaltet.

610px

Begrenzungsmethode

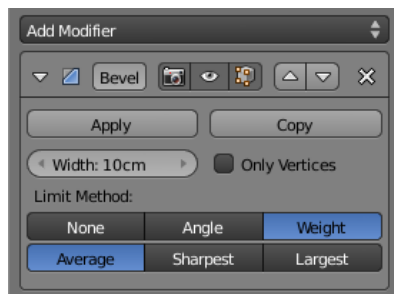
Hier kann eingestellt werden wo und wie eine Fase am Mesh angebracht wird. Die erste Reihe zeigt 3 sich gegenseitig ausschließende Algorithmen die zusätzliche Optionen haben.

"Keine"

Keine Begrenzung des Modifikators.

"Winkel"

Dieser Knopf stellt den Modifikator so ein das nur noch Kanten mit Winkeln größer als dem angegebenen mit einer Fase versehen werden.



Faseneinfluss

Einfluss

Der Faseneinfluss bestimmt wie die Fase speziell angewendet wird. Siehe [hier](#) wie man Faseneinflüsse festlegt. The three options specify what edge weight to use for weighting a vertex.

Die drei Optionen spezifizieren wie der Kanteneinfluss benutzt werden soll.

Durchschnitt

Benutzt den durchschnittlichen Einfluss am Vertex.

Kleinsten

Benutzt den kleinsten Einflusswert der anliegenden Kanten an Vertices.

Größten

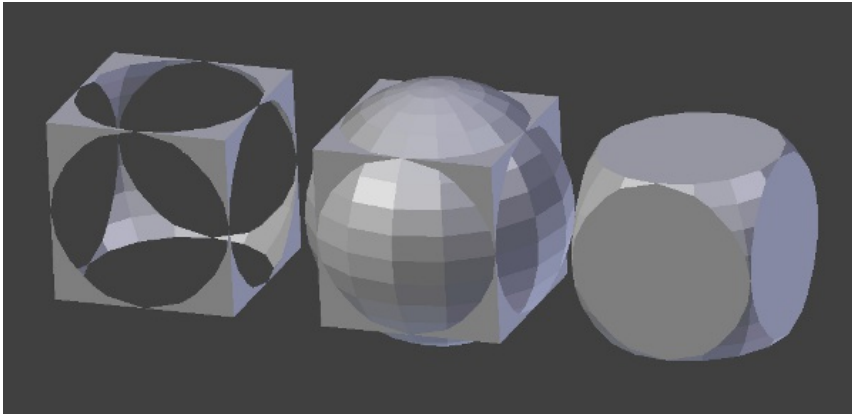
Benutzt den größten Einflusswert der anliegenden Kanten an Vertices.

Boolean Modifikator

Mode: Beliebiger Modus

Panel: Modifikator

Beschreibung



Die Differenz, die Vereinigung, und die Kreuzung zwischen einem Würfel und einer UV Kugel, mit dem auf den Würfel angewandten Modifikator

Der Boolean Modifikator verwendet eine von drei bool'schen Operationen (Difference (Negierung), Union (Trennung), und Intersect (Verbindung)), um ein einzelnes Objekt aus zwei Mesh Objekten zusammenzusetzen.

Optionen



Boolean modifier

Operation

Difference

Der Ziel-Mesh wird vom modifizierten Mesh subtrahiert (negativ).

Union

Der Ziel-Mesh wird mit dem modifizierten Mesh verbunden.

Intersect

Der Ziel-Mesh wird vom modifizierten Mesh subtrahiert.

Object

Der Name des Zielobjektes. Muss ein Mesh sein.

Siehe auch

- [Doc:Manual/Modifiers/Mesh/Booleans](http://doc.blender.org/manual/2.4/modifiers/mesh/booleans) (Dokumentation für das Blender 2.4 Boolean Mesh Editierwerkzeug inklusive einiger Beispiele in englisch)

Bauen

Modus: Objektmodus

Panel: Modifikatoren

Beschreibung

Der Bauen Modifikator bringt die Flächen des Mesh Objektes dazu nach und nach, nach einer bestimmten Zeit zu erscheinen. Wenn das Material anstelle eines Standard-Materials ein Halo ist, erscheinen die Vertices des Mesh, nicht die Flächen, nacheinander.

Standardmäßig, erscheinen die Flächen (oder Vertices) in der Reihenfolge in der sie im Speicher abgelegt worden sind (per Vorgabe, die Reihenfolge der Erstellung). Diese Reihenfolgen können im Editormodus mittels Vertices sortieren (CtrlAltF) geändert werden.

Optionen



Build modifier

Start

Der Start-Frame des Build-Prozess.

Length

Die Anzahl der Frames auf die gebaut werden soll.

Randomize

Versetzt die Reihenfolge in der die Flächen gebaut werden zufällig.

Seed

Der Zufallsgenerator. Ändern Sie dessen Optionen, um eine andere, "zufällige", Reihenfolge der Erscheinung festzulegen – Diese Reihenfolge ist immer die selbe für ein bestimmtes (Zufälligkeit, Objekt) Set.

Dezimieren - Modifikator

Beschreibung

Der Modifikator "Dezimieren" reduziert die Vertices/Seiten von einem Objekt mit einer minimalen Formänderung. Dieser Modifikator ist nicht bei Objekten, bei denen jede Ecke und Seite zur korrekten Strukturdarstellung notwendig ist anzuwenden, sondern bei einer Struktur die Resultat von komplexem Modellieren mit Vorgängen wie dem proportionellen Editieren, sukzessiven Verfeinerungen und möglicherweise einer Konvertierung von einem oberflächenunterteilungs-modifizierten in ein nicht Modifiziertes Gitter ist. Dabei endet man zumeist mit Gitterstrukturen bei denen viele Vertices nicht notwendig sind.

Der Dezimieren - Modifikator ist ein schneller und einfacher Weg die Polygonenanzahl nicht-destruktiv zu verringern. Dieser Modifikator demonstriert die Vorteile eines Gitter - Modifikationssystems, weil es zeigt wie ein Vorgang, welcher normalerweise permanent ist und das originale Gitter zerstören würde, mittels eines Modifikators interaktiv und sicher gelöst werden kann.

Anders als die Mehrzahl der vorhandenen Modifikatoren, bietet der Dezimieren - Modifikator nicht an die Änderungen im Edit Mode darzustellen.

Dezimieren arbeitet lediglich mit Dreiecken, weshalb jede viereckige Fläche in 2 Dreiecke gespalten wird um es zu dezimieren.

Optionen

Ratio (Verhältnis)

Das Verhältnis, welches nach der Dezimierung vorhanden sein soll, von 0.0(0%, bedeutet, dass alle Flächen vollständig entfernt wurden) bis 1.0(100%, das Gitter ist vollständig intakt, außer das Vierecke trianguliert wurden). Im Laufe von 1.0 zu 0.0 wird das Gitter immer weiter dezimiert bis es nicht mehr aussieht wie das originale Gitter.

Face Count (Flächenanzahl)

Dieses Feld zeigt die Anzahl von resultierenden Flächen, bei Bestätigung des Dezimieren-Modifikators.

Beispiele Einfache Fläche

Ein simples Beispiel ist eine einfache Fläche und ein nicht deformiertes 4*4 Raster. Beide sehen identisch aus, aber die einfache Fläche hat nur eine Fläche und 4 Vertices, während das Raster 9 Flächen bei 16 Vertices hat. Demzufolge existieren viele unnötige Flächen und Vertices. Der Dezimieren - Modifikator eliminiert diese unnötigen Flächen.

Dezimierter Zylinder

Wir nehmen das einfache Beispiel eines Zylinders, welcher mit seinen 32 Standard-Segmenten - dies macht 96 Flächen - dezimiert werden soll. Wenn der Dezimieren-Modifikator aktiviert wird, erhöht sich die Flächenanzahl! Das liegt an der Tatsache, dass der Modifikator alle Vierecke in jeweils 2 Dreiecke umwandelt, was steils die Flächenanzahl erhöht.

Die Hauptintention des Dezimieren-Modifikators ist es die Gittergröße durch eine Reduktion der Vertices und Flächen zu verringern, aber gleichzeitig die Form des Objektes möglichst beizubehalten.

Im folgenden Bild wird das Verhältnis von Aufnahme zu Aufnahme, von 100% auf 5%(was einem Verhältnis von 0.05 entspricht) verringert. Beachte, dass die Flächenanzahl sich von 128 auf 88, bei einem Verhältnis von 0.6, verringert hat und der Zylinder dennoch so ziemlich wie ein Zylinder aussieht, wir aber 40 Flächen gespart haben.

 ``

[Wie man sehen kann, wenn das Verhältnis 0.1 erreicht, so sieht der Zylinder eher nach einem Würfel aus. Und bei 0.05, sieht es nicht mal mehr danach aus!](#)

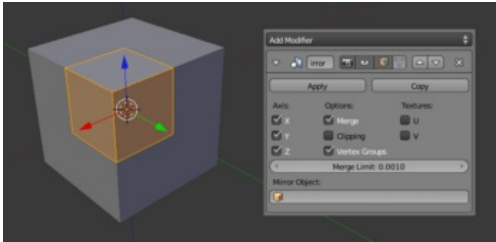
[Wenn die Flächenanzahl und das Aussehen dem Wunsch entsprechen, kann der Modifikator angewendet \(Apply\) werden. Wenn man viele Dreiecke zurück zu Vierecken konvertieren will, um noch mehr Ressourcen zu sparen, kann man in den Edit Mode wechseln, alle Vertices markieren \(A\), und Alt+J drücken.](#)

[Mirror Modifier](#)

[Modus: Any mode](#)

[Panel: Modifiers](#)

Beschreibung



Die Ecke von einem Würfel wurde an allen 3 Achsen gespiegelt und es bildet sich... natürlich... ein Würfel.

Der Mirror Modifizier spiegelt ein Mesh automatisch an seinen lokalen X, Y und/oder Z Achse. Außerdem kann auch ein anderes Objekt als Spiegelzentrum verwendet werden. Es werden dann die lokalen Achsen dieses Objekts verwendet und nicht mehr die eigenen. Es können die Vertices zusammengefügt werden, die nach der Spiegelung dicht aneinander liegen. Die Toleranz bis zu welcher Entfernung die Punkte vereint werden kann festgelegt werden. Es kann zudem verhindert werden, das einzelne Punkte auf die andere Seite der Spiegelung gehen können. Es können nicht nur das Mesh gespiegelt werden, sondern auch Vertex Groups und UV Koordinaten.

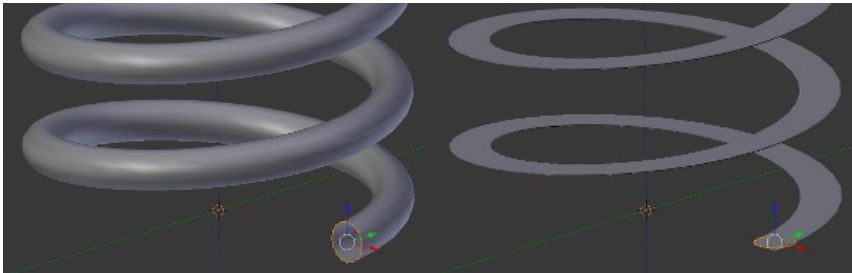
Screw Modifikator

Mode: Beliebiger Modus

Panel: Modifikator

Beschreibung

Der Screw-Modifikator ist dem Screw-Werkzeug im Tool Shelf gleich, dass aus einem Profilobjekt und einem Mesh oder einer Curve, eine Spirale erzeugt.



Das richtige Ausrichten des Profilobjektes ist wichtig.

Das Profil sollte zwei dimensional und richtig ausgerichtet zur grundsätzlichen Richtung des Objektes, aber nicht zur Spiralachse sein.

Optionen



Screw Modifikator

Axis

Die Achse, um die sich die Spirale windet.

Screw

Die Höhe einer Spiralenwiederholung.

AxisOb

Der Name eines Objektes, das die Achsenrichtung definiert.

Object Screw

Verwendet ein Axis Object, das den Wert von Screw definiert.

Angle

Gradzahl für eine einzelne Spiralendrehung.

Steps

Anzahl von Schritten, die für eine einzelne Drehung verwendet werden (angezeigt in der 3D Ansicht.)

Render Steps

Wie oben, wird während des Renderns verwendet. Durch erhöhen des Wertes wird die Qualität verbessert.

Calc Order

Die Anordnung von Kanten wird berechnet, um Probleme mit Normalen zu vermeiden. Nur erforderlich für Meshes, nicht bei Kurven.

Flip

Dreht die Normalenrichtung um.

Iterations

Anzahl der Drehungen.

Solidify Modifikator

Modus: Beliebiger Modus

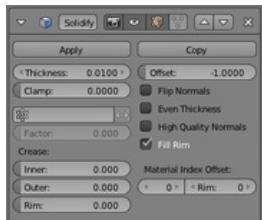
Panel: Modifikator

Beschreibung

Der Solidify Modifikator nimmt eine Oberfläche eines Meshes und fügt dieser eine Stärke hinzu.

Ursprünglich wurde er als Standalone Skript von Campbell Barton (auch bekannt als Ideasman) geschrieben, Solidify war [in Blender 2.44 enthalten](#) und letzten Endes als Transformationsmodifikator in Blender 2.5.

Optionen



Solidify Modifikator

Thickness

Die Wandstärke des Materialauftrags.

Offset

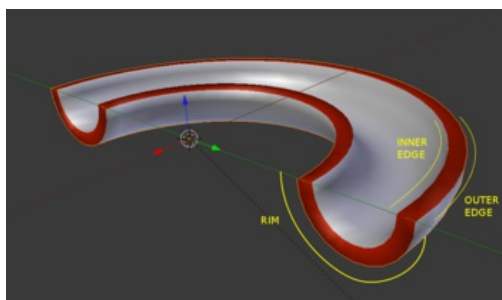
Ein Wert zwischen **-1** und **1**, um die Position des Materialauftrages zu bestimmen, innen oder außen vom original Mesh. Wenn der Wert Null ist, wird der Versatz, des Materialauftrags, mittig zum original Mesh sein.

Vertex Group

Schränkt den Modifikator auf nur diese Vertex-Gruppe ein.

Invert

Kehrt die vorherige Auswahl um.



Ränder und Kanten. In diesem Beispiel wurde dem Objekt ein zweites Material zugewiesen, um den Rand rot zu färben.

Crease

Inner

Weist eine Falte den inneren Kanten zu.

Outer

Weist eine Falte den äußeren Kanten zu.

Rim

Weist eine Falte dem Rand zu.

Even Thickness

Erhält die Dicke, durch ausrichten scharfer Kanten, aufrecht. Dies verbessert nicht nur die Qualität sondern erhöht auch die Berechnungszeit.

High Quality Normals

Normalen werden berechnet, um eine gleiche Dicke zu erzeugen. Dies verbessert nicht nur die Qualität sondern erhöht auch die Berechnungszeit.

Fill Rim

Schließt die Lücke zwischen den inneren und äußeren Rändern.

Rim Material

Verwendet das zweite Material des Objekts für den Rand, dies wird als Ausgleich vom aktuellen Material angewandt.

Hinweis

- Die Modifikator-Dicke wird vor der Objektskalierung angewandt, wenn man eine feste Dicke erhalten möchte, ist es wichtig unskalierte Objekte zu verwenden (es sei denn es ist gewollt).
- Solidify thickness is an approximation, while "Even Thickness" and "High Quality Normals", should yield good results, The architectural/CAD modeling the final wall thickness isn't guaranteed, depending on the mesh topology. To look at it differently -

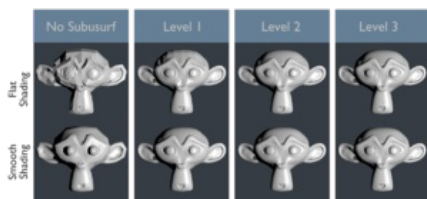
maintaining precise wall thickness in some cases would need to add / remove faces on the offset shell - something this modifier doesn't do since this would add a lot of complexity and slow down the modifier.

Oberflächenunterteilungsmodifikator

Mode: Alle Modi

Panel: Modifikatoren > Hinzufügen > Erzeugende > Oberflächenunterteilung

Hotkey: s + Ctrl0, Ctrl1, Ctrl2, Ctrl3, Ctrl4, Ctrl5. Dies beeinflusst aber nur den Vorschauwert.



null bis dreifache Oberflächenunterteilung mit und ohne weiche Schatten. Dieses Bild wurde mit dieser [SubsurfDemo.blend](#) gerendert.

Oberflächenunterteilung ist eine Methode um den Flächen eines einfachen Meshes eine glattere Form zu geben. Es erlaubt das Erzeugen von detaillierten Meshes ohne erhöhten Speicheraufwand.

Die Methode erzeugt virtuelle Geometrie die in Echtzeit erzeugt wird, die aber durch anwenden in echte umgewandelt werden kann. Wie andere Modifikatoren ist dies ein nicht destruktiver Modifikator der das zugrundeliegende Mesh nicht verändert bis der "anwenden" Knopf benutzt wird (und selbst dann kann dieser Schritt rückgängig gemacht werden)

Genau wie bei anderen Modifikatoren hat die Reihenfolge der Ausführung einen großen Einfluss auf das Ergebnis des Modifikators. Siehe [den Modifikatorstapel](#) für mehr Information.

Es gibt zwei zur Verfügung stehende Algorithmen:

- Einfach

unterteilt das Mesh wie die [Unterteilen](#) Funktion.

- Catmull-Clark (standardmäßig eingestellt)

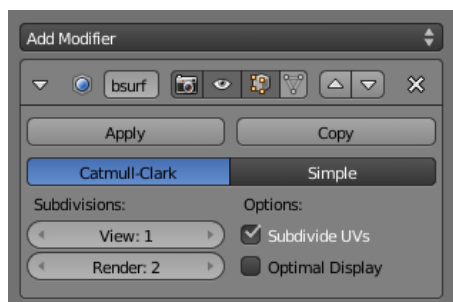
der gleichzeitig unterteilt und glättet. Siehe [Catmull-Clark](#).

Es ist wichtig daran zu denken das dies eine andere Funktion ist als [weiche Schatten](#). Der Unterschied ist im Bild rechts zu sehen.

MultiAuflösungsmodifikator

Ein anderer Weg ein Mesh zu unterteilen ist der [MultiAuflösungsmodifikator](#). Er unterscheidet sich insofern von diesem das es möglich ist das Mesh auf mehreren Unterteilungsstufen zu verändern ohne angewendet zu werden oder Information aus den anderen Stufen zu verlieren. Er ist schwerer zu benutzen, erlaubt aber eine bessere Feinsteuerung.

Optionen



Modifikatorpanel

"Oberflächenunterteilung" ist ein [Modifikator](#). Um ihn zu einem Mesh hinzuzufügen klicke auf "Hinzufügen" und wähle die "Oberflächenunterteilung" aus der Liste aus.

Typ

Erlaubt die Einstellung des Unterteilungsalgorithmus:
Catmull-Clark

Die Standardoption unterteilt und glättet Oberflächen. Siehe [\[1\]](#) für mehr Informationen.

Einfach

Unterteilt die Oberflächen nur ohne zu glätten.

Stufenanzahl

Fügt rekursiv Geometrie hinzu.
Ansicht

Stellt die Unterteilung für die 3D Ansicht ein.

Render

Stellt die Unterteilung für den Render ein.

Die richtige Kombination dieser Einstellungen erlaubt eine leichte und schnelle Annäherung an eine Form in der 3D Ansicht bei einem hohen Renderdetailgrad.



3D Ansicht und Render

Es sollte sichergestellt sein das die Unterteilung für den Render immer größer ist als für die 3D Ansicht, sonst wird die Vorschau detaillierter dargestellt als das Ergebnis.



Unterteilung von UVs an
und aus -- siehe [.blend](#)

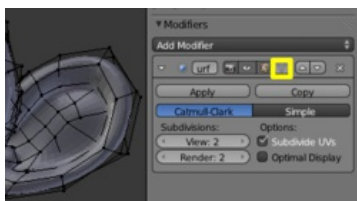
Optionen

Unterteile UVs

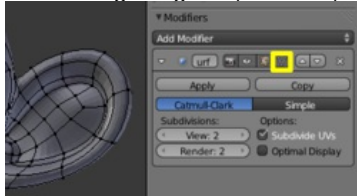
Bei Aktivierung werden neue UV Koordinaten für das unterteilte Mesh errechnet. Das bewirkt eine Verzerrung der Texturen. Siehe diese Beispieldatei: [Manual-Modifiers-Generate-Subsurf-SubdivideUVsExample.blend](#).

Optimale Darstellung

Beschränkt die Darstellung auf ein verzerrtes Gittermodell und lässt die Unterteilungsdarstellung weg.



Bearbeitungskäfig aus (Standard)



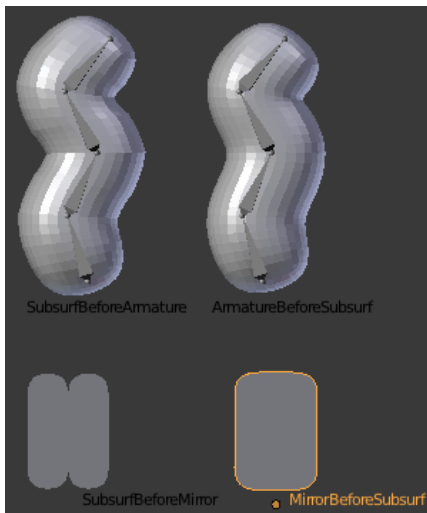
Bearbeitungskäfig an

Bearbeitungskäfigmodus

Um die Ergebnisse der Unterteilung darzustellen und zu verändern während das Mesh bearbeitet wird, muss der Bearbeitungskäfig eingeschaltet werden. Dies bewirkt das die Vertices in ihrer modifizierten Position dargestellt werden und von dort verändert werden können.

Beachte den Vergleich rechts. Mit dem Bearbeitungskäfig aus sind einige Vertices unter dem Mesh vergraben. Die Bearbeitungskäfig Version des Meshes hat zwar nicht diesen Nachteil allerdings kann das Mesh so zu gut aussehen. Wie eckig Suzannes Ohr ist, ist nur in der normalen Ansicht sichtbar! Es ist also ratsam den Bearbeitungskäfig nur selten einzusetzen.

Ausführungsreihenfolge



Beachte das der Armaturmodifikator vor dem Unterteilungsmodifikator ebenso bessere Ergebnisse liefert wie der Spiegelungs- vor dem Unterteilungsmodifikator.

Die [Ausführungsreihenfolge](#) ist häufig sehr wichtig. Siehe [Manual-Modifiers-Generate-Subsurf_OrderOfExecution.blend](#) für die Quelle der Screenshots.

Kontrolle

Die Unterteilung rundet Ecken ab, was häufig nicht erwünscht ist. Es gibt mehrere Lösungen:

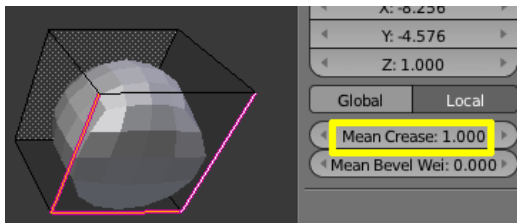
gewichtete Kanten

Mode: Bearbeitungsmodus (Mesh)

Panel: 3D View → Transformseigenschaften

Hotkey: ⇧ ShiftE

Menu: Mesh → Kanten → Kanten gewichten



Ein unterteilter Würfel mit gewichteten Kanten.

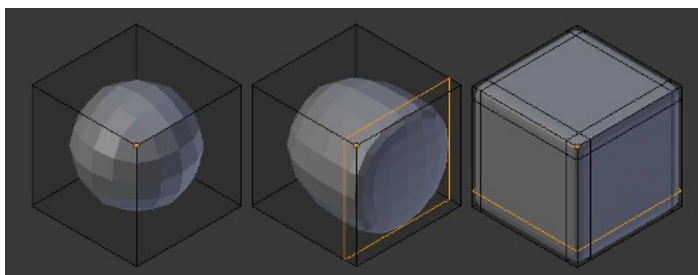
Gewichtete Kanten erlauben eine Einstellung der Kanten die nicht verzerrt werden sollen. Das erlaubt das Einstellen von glatten oder scharfen Kanten.

Die Gewichtung der Kanten kann unter den Transformationseigenschaften geändert werden.

Kantenschleifen

Mode: Bearbeitungsmodus (Mesh)

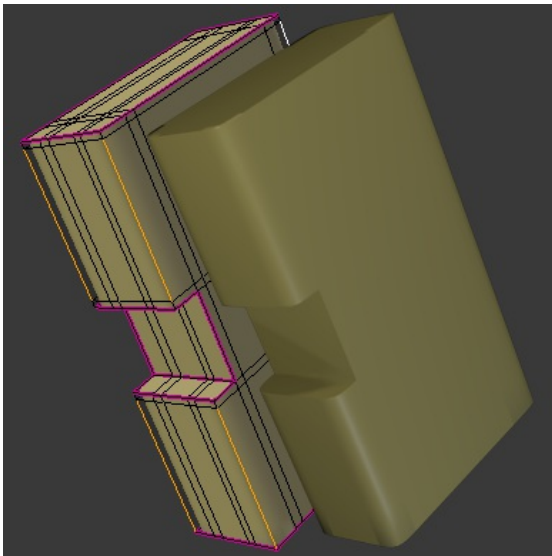
Hotkey: CtrlR



Eine 2 fache Unterteilung eines Würfels mit einer und mit 6 Kantenschleifen.

Der Unterteilungsmodifikator zeigt sehr gut warum eine saubere Topologie sehr wichtig ist. Wie man im Bild sehen kann hat der Unterteilungsmodifikator einen starken Einfluss auf das Mesh. Ein Mesh mit guten Flächenschleifen erlaubt das einfache einsetzen oder gewichten von Kanten die einen entsprechenden Einfluss auf das Aussehen des Meshes haben.

Kombination

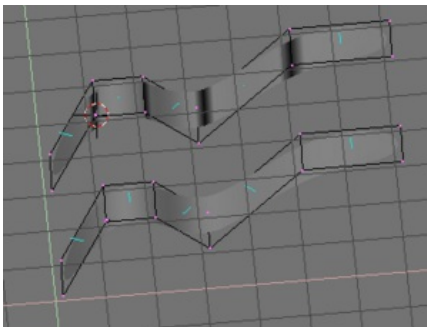


Lila Kanten sind gewichtet, Orange werden abgerundet.
Siehe: <File:WoodBlock.blend>

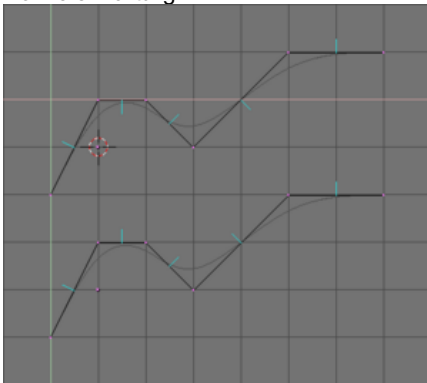
Es ist gut über "glatte, harte Schatten", "Kantengewichtung" und "Kantenschleifen" bescheid zu wissen. Zum Beispiel in einem Holzblock das eine Nut hat. Die von der Fabrik abgeschliffenen Kanten sind rund, aber wo die Säge den Block zersägt hat gibt es harte Kanten.

Grenzen & Hilfen

Blenders Unterteilungssystem erzeugt gut geglättete Meshes aber jede erzeugte Fläche hat die Flächennormale der Fläche aus der sie erzeugt wurde.



Solide Ansicht von unterteilten Meshes mit inkonsistenten Normalen (oben) und konsistenten Normalen (unten). Die dunklen Flächen entstehen aufgrund der Normalenrichtung.

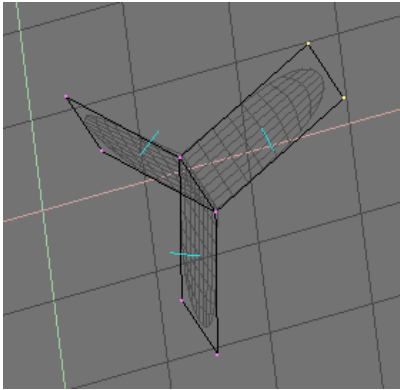


Seitliche Ansicht der oben gezeigten Bilder mit zufälligen Normalen (oben) und kohärenten Normalen (unten).

Entgegengesetzte Normalen können schwarze Flächen erzeugen.

Um das zu beheben können entweder die Normalen Neuberechnet werden CtrlN oder durch manuelles auswählen der Flächen und umkehren der Normalen.

Beachte das Blender Normalen nicht bei Meshes Neuberechnen kann die an einer Kante mehr als zwei Flächen haben. Es gibt dann kein einheitliches "aussen" dem die Normalen zugewandt werden könnten.



Eine Mesh ohne definiertes "aussen".

Leistung

Eine starke Unterteilung erfordert eine große Menge Speicherplatz und eine schnelle Grafikkarte. Blender kann crashen falls die Unterteilung zu groß wird und die Systemleistung übersteigt.

In dieser Tabelle sind einige Vertex, Kanten und Flächenzahlen zur Verdeutlichung aufgeführt:

Würfel Unterteilungsstufe	Resultierende Verts	Resultierende Kanten	Resultierende Flächen
0	8	12	6
1	26	48	24
2	98	192	96
3	386	768	384
4	1538	3072	1536
5	6146	12288	6144
6	24578	49152	24576
Formel	$3 \cdot 2^{n+1} + 4$	$3 \cdot 4^n$	$6 \cdot 2^n$

Hier ist eine Tabelle für ähnliches Verhalten einer Ebene:

Ebenenunterteilungsstufe	Resultierende Verts	Resultierende Kanten	Resultierende Flächen
0	4	4	1
1	9	12	4
2	25	40	16
3	81	144	64
4	289	544	256
5	1089	2112	1024
6	4225	8320	4096
Formel	$(2^{n+1} + 2)^2 / 4$	$2^{n+1} \cdot (2^{n+1} + 2)$	4^n

Und für Dreiecke:

Dreieckunterteilungsstufe	Resultierende Verts	Resultierende Edges	Resultierende Faces
0	3	3	1
1	7	9	3
2	19	30	12
3	61	108	48
4	217	408	192
5	817	1584	768
6	3169	6240	3072
Formel	$3 \cdot 2^{n+1} + 3$	$3 \cdot 2^{n+1} \cdot (2^{n+1} + 1)$	$3 \cdot 2^{2n}$ [n=0 gerundet]

Category:Modifiers]]

Skelettmodifikator

Mode: Object mode

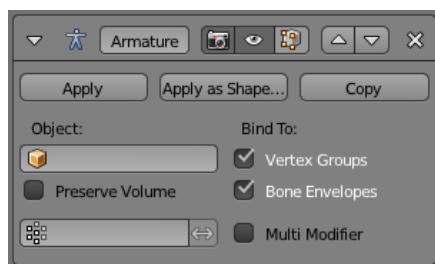
Panel: Modifiers

Beschreibung

Der Skelettmodifikator wird zum erstellen von Skeletten zum Animieren benutzt. Es können mit Vertexgruppen und mehreren Modifikatoren können mehrere Skelette dazu benutzt werden ein einziges Objekt zu animieren.

Das Benutzen eines Skelettsystems erlaubt das Verformen des Objekts so das die nicht per Hand vorgenommen werden muss. Der Skelettmodifikator erlaubt es Objekte einfach durch auswahl des gewünschten Skeletts zu animieren ohne das Unterordnungssystem zu benutzen. Siehe [das Kapitel über das Aufbauen von Skeletten und deren Animation](#).

Optionen



Skelettmodifikator

Object

Name des Skelettobjekts das benutzt werden soll.

Erhalte Volumen

Benutzt Quaternions für eine weichere Rotation.

Vertexgruppe

Der Name der Vertexgruppe deren Gewichtung festlegt wie stark dieser Modifikator die Vertexgruppe beeinflussen soll.

MultiModifikator

Benutzt die gleichen Daten wie der vorhergehende Modifikator und erlaubt das Benutzen von mehreren Armaturen an einem Objekt. Das hat den Vorteil das die Verzerrung durch die Skelette parallel ermittelt und dann verrechnet werden, im Gegensatz zur Berechnung hintereinander und Addition der Werte.

Einflussoptionen

Das Skelett kann Einfluss haben auf:

Vertex Gruppen

Schaltet ein oder aus ob das Skelett Vertex Gruppen zum Deformieren benutzt.

Knochen Umgebung

Schaltet ein oder aus ob Knochen ihre Umgebung zur Deformation benutzen.

Umkehrung

Kehrt den Einfluss der Vertexgruppe um.

Ausgießen Modifikator

Panel: Modifiers

Beschreibung

Dieser Modifikator verschiebt die Form eines Mesh, einer Kurve, einer Oberfläche, oder eines Gitters zu einen der wenig vordefinierten Formen (Kugel, Zylinder, Quader).

Es ist äquivalent zum Zu Kugel Werkzeug im Bearbeitungs Kontext (Mesh → Transformieren → Zu Kugel Alt+ShiftS) und was andere Programme "Kugelförmig formen" oder "Wölben" nennen, aber, wie oben erwähnt ist es nicht auf das Ausgießen einer Kugel beschränkt.

Tipp

Der [Weich Modifikator](#) ist ein guter Begleiter zum Ausgießen, da die ausgegossene Form manchmal Glätten erfordert um besser auszusehen oder auch nur zum Ausbessern der Shading Artefakte.

Wichtig

Aus Performance Gründen, funktioniert dieser Modifikator nur mit lokalen Koordinaten. Wenn das modifizierte Objekt falsch aussieht, müssen Sie wohl die Rotation des Objektes anwenden (StrgA, speziell wenn zu einem Zylinder ausgegossen wird).

Optionen



cast modifier

Cast Type

Menü zum Auswählen des Ausguss Types (Zielform): Kugel, Zylinder oder Quader.

X, Y, Z

Wechsel Knöpfe zum Aktivieren/Deaktivieren des Modifikators in den X, Y, Z Achsen Richtungen (X und Y nur für Zylinder Ausguss Typ)

Faktor

The factor to control blending between original and cast vertex positions. It's a linear interpolation: **0.0** gives original coordinates (aka modifier has no effect), **1.0** casts to the target shape. Values below or above [0.0, 1.0] deform the mesh, sometimes in interesting ways.

Radius

If non-zero, this radius defines a sphere of influence. Vertices outside it are not affected by the modifier.

Größe

Alternative size for the projected shape. If zero, it is defined by the initial shape and the control object, if any.

Von Radius

If activated, calculate Size from Radius, for smoother results.

Vertex Gruppe

A vertex group name, to restrict the effect to the vertices in it only. This allows for selective, realtime casting, by painting vertex weights.

Kontrollobjekt

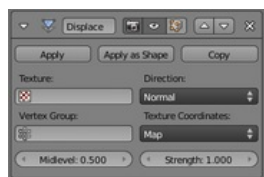
Der Name eines Objektes der den Effekt kontrolliert. Der Ort dieses Objektes definiert das Zentrum der Projektion. Ebenso, transformieren dessen Größe und Rotation die projizierten Eckpunkte. Tipp: Animierung (mit Schlüsselbildern) dieses Kontrollobjektes animiert auch das modifizierte Objekt.

Verschiebungen Modifikator

Beschreibung

Der Verschieben Modifikator verdrängt Knoten in einem Netz basierend auf der Intensität einer Textur. Es können entweder prozedurale, oder Bild Texturen verwendet werden. Die Verschiebung kann entlang einer bestimmten lokalen Achse, der normalen Vertex Achse verlaufen, oder es es können separate RGB Komponenten der Textur genutzt werden um Eckpunkte entlang der lokalen X, Y oder Z Richtungen zu verschieben.

Optionen



Displace modifier

Textur

The name of the texture from which the displacement for each vertex is derived.
If this field is empty, the modifier will be disabled.

Vertex Gruppe

The name of a vertex group which is used to control the influence of the modifier.
If VGroup is empty, the modifier affects all vertices equally.

Mittel Level

The texture value which will be treated as no displacement by the modifier. Texture values below this value will result in negative displacement along the selected direction, while texture values above this value will result in positive displacement. This is achieved by the equation $(\text{displacement}) = (\text{texture value}) - \text{Midlevel}$.

Recall that color/luminosity values are typically between **0.0** and **1.0** in Blender, and not between **0** and **255**.

Richtung

Die Richtung wo die Eckpunkte entlang verschoben werden sollen.
Kann eine der folgenden sein:

- X – Verschiebung entlang der lokalen X Achse.
- Y – Verschiebung entlang der lokalen Y Achse.
- Z – Verschiebung entlang der lokalen Z Achse.
- RGB -> XYZ – Verschiebung entlang der lokalen XYZ Achsen individuell durch Nutzung von RGB Komponenten der Textur.
- Normal – Verschiebung entlang der normalen Vertex.

Textur Koordinaten

The texture coordinate system to use when retrieving values from the texture for each vertex.
Can be one of the following:

- UV – take texture coordinates from face UV coordinates.

UV Layer

The UV coordinate layer from which to take texture coordinates.

If the object has no UV coordinates, it uses the Local coordinate system. If this field is blank, but there is an UV coordinate layer available (e.g. just after adding the first UV layer to the mesh), it will be overwritten with the currently active UV layer.

Note

Since UV coordinates are specified per face, the UV texture coordinate system currently determines the UV coordinate for each vertex from the first face encountered which uses that vertex; any other faces using that vertex are ignored. This may lead to artifacts if the mesh has non-contiguous UV coordinates.

- Object – take the texture coordinates from another object's coordinate system (specified by the Object field).

Object

The object from which to take texture coordinates. Moving the object will therefore alter the coordinates of the texture mapping. Take note that moving the original object will **also** result in a texture coordinate update. As such, if you need to maintain a displacement coordinate system while moving the object to which the displacement is set, you will also have to move the related object at the same rate and direction.

If this field is blank, the Local coordinate system is used.

- Global – Holt sich die Textur Koordinaten von dem globalem Koordinaten System .
- Lokal – Holt sich die Textur Koordinaten von dem lokalen Koordinaten System des Objekts.

Stärke

Die Stärke der Verschiebung. Nach Verrechnung des Mittel Level Wertes, wird die Verschiebung mit dem Wert der Stärke multipliziert um die finale Vertex Position zu liefern. Dies wird durch die folgende Gleichung erreicht $(\text{Vertex Position}) = (\text{Verschiebung}) \times \text{Stärke}$.


Ein negativer Stärke Wert kann zum Invertieren des Modifikator Effektes genutzt werden.

Siehe

- Blender Artists Beitrag: [Displace modifier tutorial](#) (September 2006)

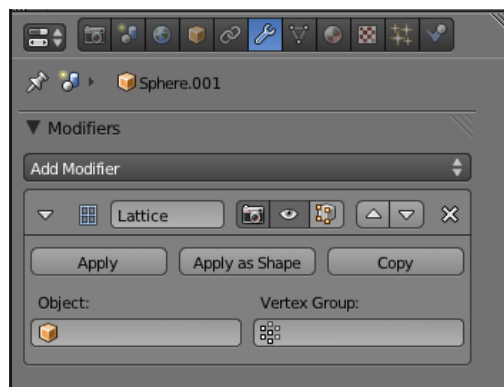
Lattice Modifier

Modus: Object Mode

Panel: Properties Window -> Kontext-Icon Modifiers 

Der Lattice-Modifier verformt das Basis-Objekt entsprechend der Form eines umgebenden Lattice (Dt.: Gitter).

Optionen



Lattice Modifier.


Object (Objekt)

Das Lattice-Objekt, mit dem das Basismeshes verformt wird.

Vertex Group (Vertex-Gruppe)

Ein optionaler Vertex-Gruppenname, durch dessen Angabe der Modifier-Effekt auf nun einen Teil des Basismeshes eingeschränkt wird.

Hinweise

Im Object Data-Kontext des Properties Windows kann man die Objektattribute des Lattice-Objekts  steuern.

Ein Lattice besteht aus einem nicht-renderbaren dreidimensionalen Gitter von Steuervertices. Sein Hauptzweck ist es, zusätzliche Verformungen auf das von ihm kontrollierte Objekt zu ermöglichen, sei es durch einen Modifier oder als Mutterobjekt. Die Zielobjekte können Meshes, Kurven, Flächen, Text, Gitter oder sogar Partikel sein.

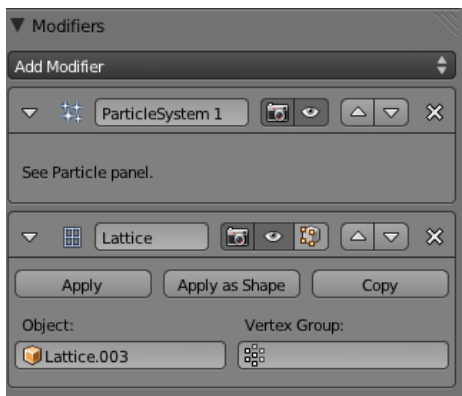
Warum sollte man ein Lattice nutzen, um ein Mesh zu verformen, statt das Mesh direkt im Edit Mode zu bearbeiten? Dafür gibt es eine Reihe guter Gründe:

- Es ist einfacher! Da ein Mesh Unmengen von Vertices enthalten kann, wäre deren einzelnes Skalieren und Verschieben Schwerstarbeit. Stattdessen reduziert sich diese Arbeit mit einem guten einfachen Lattice auf die Verschiebung nur weniger Steuervertices.
- Es ist schöner! Die Verformung sieht wesentlich besser aus!
- Es ist schneller! Man kann dasselbe Gitter zur Verformung mehrerer Meshes benutzen. Hierzu ordnet man einfach jedem Mesh jeweils einen Lattice Modifier zu, der immer auf dasselbe Lattice zugreift.
- Es ist gute Praxis! Ein Lattice kann verwendet werden, um mit minimaler zusätzlicher Arbeit und geringem Ressourceneinsatz unterschiedliche Versionen eines Meshes zu erzeugen. Dies führt zu einem optimalen Scene-Design und minimiert den Modellierungsaufwand. Ein Lattice verändert nicht die Textur-Koordinaten einer Meshoberfläche. Feine Änderungen an Mesh-Objekten können so einfach erreicht werden, ohne das Mesh selbst zu verändern.

Beispiele/Tutorials

Es gibt Beispiel-Tutorials für Version 2.4 im [Tutorials](#)-Abschnitt. Ein 2.6-Tutorial zeigt das [Formen einer Gabel](#).

Partikel und Lattice



Partikel folgen einem Lattice.

Partikel folgen einem Lattice, wenn die Modifier-Reihenfolge richtig eingestellt ist: Erst die Partikel, dann das Gitter!

Einleitung in die Beleuchtung

Beleuchtung ist ein sehr wichtiges Thema beim Rendern, Modellieren sowie beim Umgang mit Materialien und Texturen. Eine noch so genau modellierte und texturierte Szene sieht ohne die richtige Beleuchtung unbefriedigend aus, während selbst ein einfaches Modell sehr realistisch aussehen kann, wenn es geschickt beleuchtet wird. Beleuchtung wird häufig vom unerfahrenen Anwender übersehen, der allgemein glaubt, dass es wie in der Realität ausreicht, die Szene durch ein einzelnes Licht zu beleuchten (wie eine Lampe, die Sonne, usw.). Das ist falsch, weil in der realen Welt - selbst wenn nur eine einzelne Lichtquelle vorhanden ist - das Licht bei Auftreffen auf Gegenstände abgestrahlt wird sowie überall in der Szene weiche Schatten wirft und nicht etwa tiefschwarze Schatten erzeugt.

Bestimmende Faktoren beim Sehen

Die Wahrnehmung der Farbe eines Objektes und der Beleuchtung der Szene wird bestimmt durch

- die individuelle Fähigkeit, verschiedene Farben sehen zu können (es kann z.B. eine partielle Farbenblindheit vorliegen),
- das Medium, in dem Sie das Bild ansehen (z.B. eine LCD-Anzeige oder bedrucktes Kunstdruckpapier),
- die Qualität der Bilder (z.B. ein JPEG von einer **40%** Komprimierung oder einer **100%** Komprimierung),
- die Umgebung, in der man das Bild betrachtet (z.B. ein CRT-Monitor mit dem grellen Schein in einem dunklen Zimmer oder in einem sonnigen blauen Zimmer) und
- die [Wahrnehmung](#) der Farbe und Intensität sowohl der Objekte als auch der Welthintergrundfarbe.

Genau das gleiche Bild, das von der Person A am Monitor B im Zimmer C betrachtet wird, kann also für Person D, die einen Ausdruck E vom Bild sieht, während sie in der U-Bahn F steht, durchaus anders aussehen.

Globale Einflüsse

In Blender sind hier die Werkzeuge zusammenfasst, die vom Anwender eingestellt werden können, um die Ausleuchtung einzurichten:

- Die Farbe der Welt: [Umgebungsbeleuchtung](#).
- Die Nutzung von [Ambient Occlusion](#) (Umgebungsverdeckung) als ein Weg, das Auftreffen von Umgebungslicht auf das Objekt darzustellen.
- Die Farbtemperatur, mit der das Licht das Beleuchtungslicht die [Material](#)-Farbe des Objektes beeinflusst.
- Der Einsatz von [Indirect Lightning](#) (indirektes Licht), mit dem Farbreflexionen eines Objektes auf ein anderes dargestellt werden.
- Die Render Engine (Blender Intern bzw. [Yafray](#)).
- Die [Lampen](#) in der Szene.

Die physikalischen Eigenschaften reflektierten Lichts in der realen Welt werden durch Ambient Occlusion (eine World-Einstellung), Buffer Shadows (welche eine Annäherung an den Schattenwurf von Objekten darstellen) sowie Ray Tracing (wodurch der Lichtpfad von Photonen ausgehend von einer Lichtquelle nachgezeichnet wird). Außerdem kann man in Blender [Indirect Lighting](#) einsetzen. Ray Tracing, Ambient Occlusion und Indirect Lighting sind rechenintensive Prozesse. Blender kann ein wesentlich schnelleres Rendern mittels seines internen Scan Line Renderers ausführen, der wirklich ein sehr guter Renderer ist. Diese Art von Render-Maschine arbeitet viel schneller, weil sie nicht versucht, das reale Verhalten des Lichts nachzubilden, sondern sich auf eine Vielzahl von vereinfachenden Rechenmodellen stützt.

Beleuchtungseinstellungen

Erst nach den oben geschilderten globalen Einflüssen sollte man sich mit dem Hinzufügen zusätzlichen Lampenlichts befassen. Hauptsächlich stehen einem die folgenden Einstellungen zur Verfügung:

- Die Art der Lichtquelle (Sun (Sonne), Spot (Punktstrahler), Lamp (Lampe), Hemi (Halbkugel) etc.).
- Lichtfarbe.
- Position und Richtung der Lichtquelle.
- Einstellungen für jede dieser genannten Lichtquellen einschließlich Energy (Lichtstärke) und Falloff (Schattenfall).

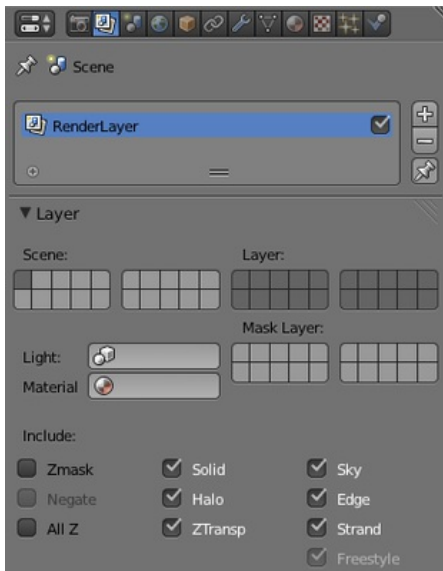
Dies führt zurück zu der Frage, wie der [Shader](#) des Materials auf Licht reagiert.

Dieses Kapitel versucht, auf die oben genannten Punkte näher einzugehen, wobei auch das Zusammenwirken von Lichtarrangements zur Ausleuchtung der Szene mitbehandelt werden wird. Im folgenden werden wir die unterschiedlichen Lichttypen und ihr Verhalten mit ihren Stärken und Schwächen analysieren. Dabei werden wir auch viele Lichtarrangements einschließlich der immer wieder gern verwendeten Dreipunkt-Ausleuchtung vorstellen.

Beleuchtung im Arbeitsprozess

In diesem Benutzerhandbuch haben wir die Ausleuchtung vor dem Material-Thema angeordnet; man sollte sein Lichtarrangement eingerichtet haben, bevor man seinen Meshes Materialien zuordnet. Da die Material Shaders auf Licht reagieren, würden diese ohne ordentliche Ausleuchtung unbefriedigend dargestellt werden. Es könnte dazu führen, die Schuld beim Shader zu suchen, obwohl es eigentlich nur die Art der Ausleuchtung ist, die einem diesen Verdross bereitet. Alle Beispielbilder dieses Abschnitts benutzen keinerlei Materialeinstellungen auf dem Ball, dem Würfel oder für den Hintergrund.

Ausblenden von Materialien zum Einrichten der Ausleuchtung



Material-Feld auf dem Render Layers-Panel

Denjenigen, die sich bereits mit dem Zuordnen von Materialien befasst hatten und nun mit der Ausleuchtung herumtüteln, raten wir, sich ein neutralgraues Standard-Material anzulegen: Keine Vertex Color, keine Face Texture, kein Shadeless, lediglich ein ganz schlichter Grauton mit den RGB-Werten (**0.8, 0.8, 0.8**).

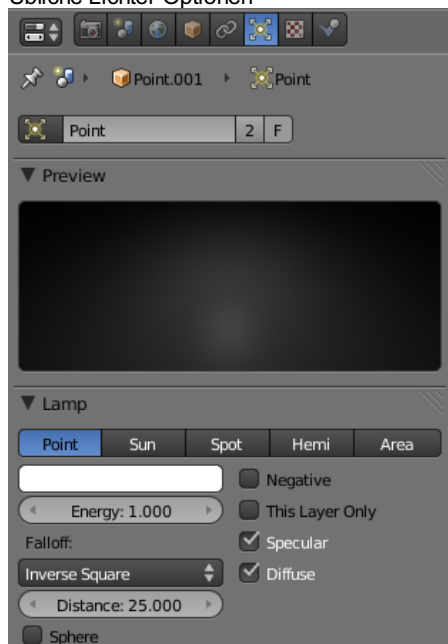
Danach geht man in das Render-Menü. Auf dem Render Layers-Panel wählt man "Grey" im Material-Feld. Dadurch werden alle möglicherweise zuvor angelegten Materialien ausgeblendet und alles in dieser neutral-tristen Farbe gerendert. Nach Aktivierung dieses Materials kann man sich nun an das Einrichten der Ausleuchtung machen. Um zu den eigentlich zugeordneten Materialien zurückzukehren, muss man einfach nur dieses Feld wieder leeren.

Lichter

Wie vorher bereits gesagt, gibt es zahlreiche Arten von Beleuchtung in Blender wie z.B. indirektes Licht oder Umgebungslicht. Die am häufigsten verwendeten sind indessen "Lichter" oder "Lampen". In diesem Abschnitt geht es um allgemeine Informationen und Einstellungen für diese Lichtquellen (mehr über lampenspezifische Details im Kapitel [Lampen](#)):

- [Lichteinstellungen](#) – übliche Einstellungen für alle Lampen.
- [Lichtdämpfung](#).
- [Texturen](#) – Hinzufügen von Texturen zu Lampen.
- [Was das Licht beeinflusst](#).
- [Lampen bezogene Einstellungen](#) – Einstellung für Lampen in anderen Zusammenhängen.

Übliche Lichter-Optionen



Lampeneigenschaften Panel

Es gibt in Blender fünf Typen von Lampen. Sie haben alle oder einige der folgenden Optionen gemeinsam:

Objekt Daten

Browse Light Object Data

Klicken Sie hier um alle Lichter in der aktuellen Szene zu sehen.

Name

Der Name der aktuell ausgewählten Licht-Objektdaten. Bearbeiten Sie hier den Namen.

Number of Users

Die Anzahl an Licht-Objekten, die gemeinsame Licht-Objektdaten besitzen.

F

Erstellt einen unechten Benutzer für diese Objektdaten.

Preview (Übersicht)

Eine Übersicht der Lichteinstellungen.

Lamp (Lampen)

Distance

Das Distance Feld zeigt die Anzahl von Blender-Einheiten (BE) an, an der die Intensität der aktuellen Lichtquelle die Hälfte seiner Leuchtkraft verliert. Objekte mit einer geringeren Anzahl von BE's werden heller, während Objekte weiter weg weniger Licht erhalten. Bestimmte Einstellungen und Licht-Abnahmetypen haben Auswirkungen, wie das Distance Feld interpretiert wird, dass bedeutend es regiert nicht immer gleich. Mehr dazu erfahren auf der Seite [light falloff](#).

Energy

Die Intensität der Lichtquelle (von **0.0** bis **10.0**).

Color

Die Farbe der Lichtquelle. Öffnet den Farbwähler.

Negative

Lässt die Lampe negatives Licht werfen.

This Layer Only

Die Lampe beleuchtet nur Objekte auf derselben Ebene, auf der sich die Lampe befindet.

Specular

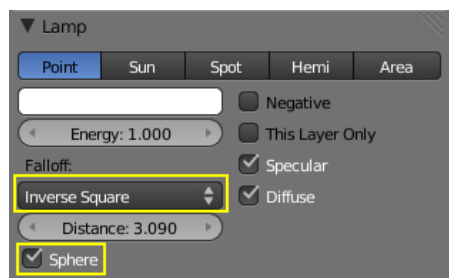
Lampe erzeugt spiegelnde Glanzlichter.

Diffuse

Die Lampe erzeugt eine diffuse Schattierung.

Lichtabschwächung

Beschreibung



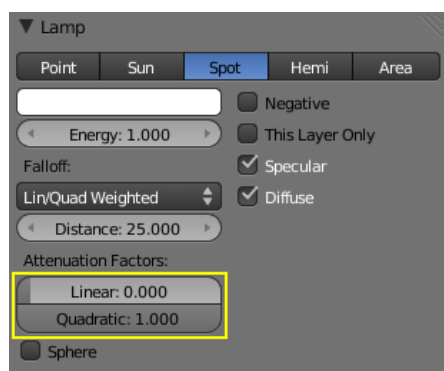
Lampe Panel, Optionen zur Abschwächung hervorgehoben

Es gibt zwei Hauptsteuerungen für Lichtabschwächung für Punkt und Scheinwerfer Lampen.

- Die Dropdown Liste zur Lichtabschwächung, und
- Der Kugel Knopf.

Typen der Abschwächung

Lin/Quad Weighted



Lampe Panel mit hervorgehobenen Lin/Quad Weighted Falloff Optionen

Wenn diese Einstellung gewählt wird, werden die zwei Slider Linear und Quadratic eingeblendet, welche die "linearen" und "quadratischen" Eigenschaften der Kurve der Abschwächung beeinflussen.

This lamp falloff type is in effect allowing the mixing of the two light attenuation profiles (linear and quadratic attenuation types).

Linear

This slider input field can have a value between **0.0** and **1.0**. A value of **1.0** in the Linear field and **0.0** in the Quadratic field in effect means that the light from this source is completely linear. This means that at the number of Blender Units distance specified in the Distance field, this light source's intensity will be half the value it was originally.

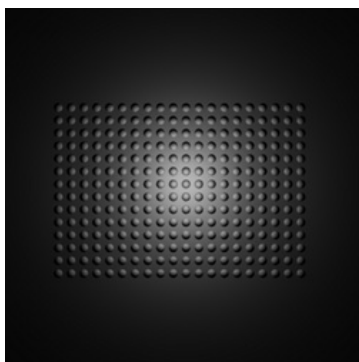
When the Quadratic slider is set to **0.0**, the formula for working out the attenuation at a particular range for full linear attenuation is:

$$I = E \times (D / (D + L \times r))$$

Where

- I is the calculated Intensity of light.
- E is the current Energy slider setting.
- D is the current setting of the Dist field.
- L is the current setting of the Linear slider.
- r is the distance from the lamp where the light intensity gets measured.

Quadratic



Lamp with Lin/Quad Weighted falloff default settings

This slider input field can have a value between **0.0** and **1.0**. A value of **1.0** in the Quadratic field and **0.0** in the Linear field means that the light from this source is completely quadratic.

Quadratic attenuation type lighting is considered a more accurate representation of how light attenuates (in the real world). In fact, fully quadratic attenuation is selected by default for Lin/Quad Weighted lamp falloff (see *Lamp with Lin/Quad Weighted falloff default settings*).

Here again, the light intensity is half when it reaches the Distance value from the lamp. Comparing the quadratic falloff to the linear falloff, the intensity decays much slower at distances lower than the set Distance, but it attenuates much quicker after Distance is reached.

When the Linear slider is set to **0.0**, the formula for working out the attenuation at a particular range for full quadratic attenuation is:

$$I = E \times (D^2 / (D^2 + Q \times r^2))$$

Where

- I is the calculated Intensity of light.
- E is the current Energy slider setting.
- D is the current setting of the Dist field.
- Q is the current setting of the Quad slider.
- r is the distance from the lamp where the light intensity gets measured.

Mixing “Linear” and “Quad”

If both the Linear and Quad slider fields have values greater than **0.0**, then the formula used to calculate the light attenuation profile changes to this:

$$I = E \times (D / (D + L \times r)) \times (D^2 / (D^2 + Q \times r^2))$$

Where

- I is the calculated Intensity of light.
- E is the current Energy slider setting.
- D is the current setting of the Dist field.
- L is the current setting of the Linear slider.
- Q is the current setting of the Quad slider.
- r is the distance from the lamp where the light intensity gets measured.

Zeroing both “Linear” and “Quad”

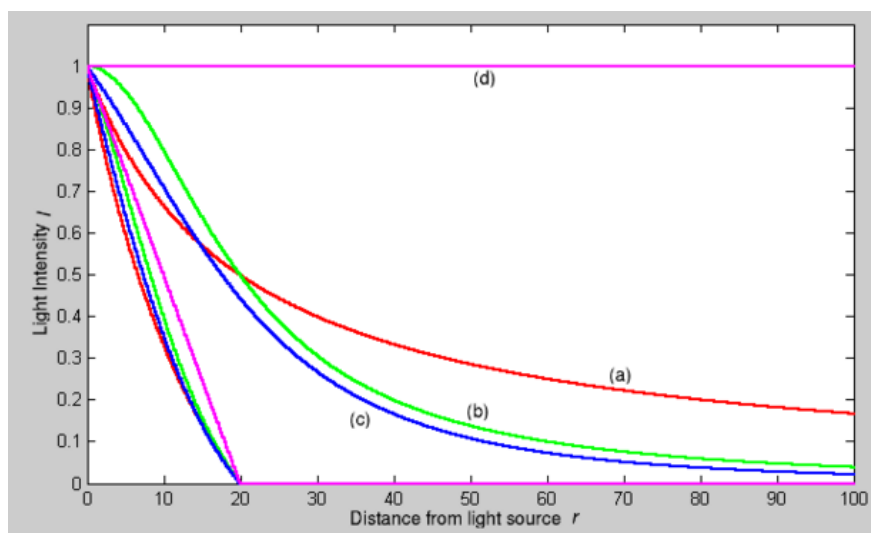
If both the Linear and Quadratic sliders have **0.0** as their values, the light intensity will not attenuate with distance. This does not mean that the light will not get darker—it will, but only because the energy the light has is spread out over a wider and wider distance. The total amount of energy in the spread-out light will remain the same, though. The light angle also affects the amount of light you see. It is in fact the behavior of light in the deep space vacuum.

If what you want is a light source that doesn't attenuate and gives the same amount of light intensity to each area it hits, you need a light with properties like the Constant lamp Falloff type.

Also, when the Linear and Quad sliders are both **0.0** values the Distance field ceases to have any influence on the light attenuation, as shown by the equation above.

Graphical Summary

Below is a graph summarizing the lin/quad attenuation type, showing attenuation with or without the Sphere option (described later).



Light Attenuation:

a) Linear (Linear=**1.0**, Quad=**0.0**); b) Quadratic (Linear=**0.0**, Quad=**1.0**);

c) Linear and quadratic (Linear=Quad=**0.5**); d) Null (Linear=Quad=**0.0**).

Also shown in the graph the "same" curves, in the same colors, but with the Sphere button turned on.

Custom Curve

The Custom Curve lamp Falloff type is very flexible.

Most other lamp falloff types work by having their light intensity start at its maximum (when nearest to the light source) and then with some predetermined pattern decrease their light intensity when the distance from the light source increases.

When using the Custom Curve Lamp Falloff type, a new panel is created called Falloff Curve. This Falloff Curve profile graph allows the user to alter how intense light is at a particular point along a light's attenuation profile (i.e. at a specific distance from the light source).



The Falloff Curve profile graph has two axes, the "Distance" axis and the "Intensity" axis.

Distance axis

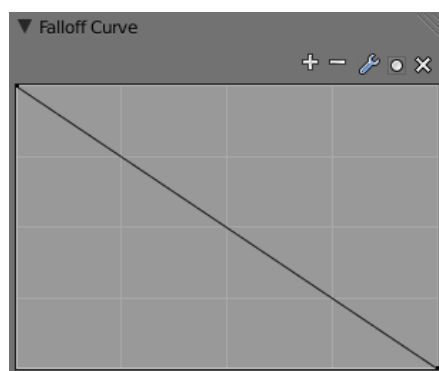
It represents the position at a particular point along a light source's attenuation path. The far left is at the position of the light source and the far right is the place where the light source's influence would normally be completely attenuated. I say "normally would" because the Falloff Curve can be altered to do the exact opposite if required.

Intensity axis

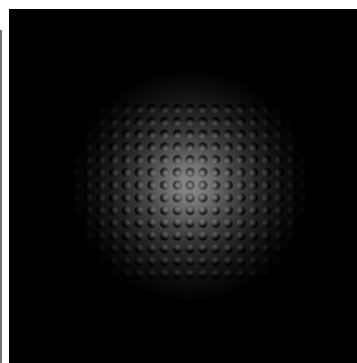
It represents the intensity at a particular point along a light source's attenuation path. Higher intensity is represented by being higher up the intensity axis, while lower intensity light is represented by being lower down on the intensity axis.

Altering the Falloff Curve profile graph is easy. Just LMB  click on a part of the graph you want to alter and drag it where you want it to be. If when you click you are over or near one of the tiny black square handles, it will turn white, indicating that this handle is now selected, and you will be able to drag it to a new position. If when you click on the graph you are not near a handle, one will be created at the point that you clicked, which you can then drag where you wish. You can also create handles at specific parts of the graph, clicking with LMB  while holding Ctrl key; it will create a new handle at the point you have clicked.

In the example below (the default for the Falloff Curve Profile Graph), the graph shows that the intensity of the light starts off at its maximum (when near the light), and linearly attenuates as it moves to the right (further away from the light source).

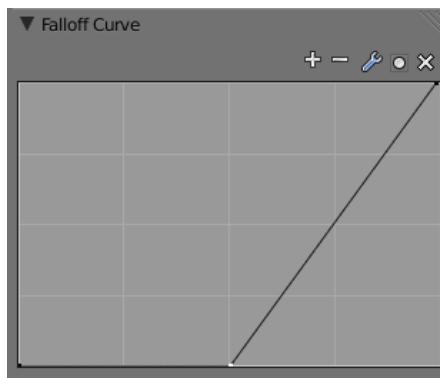


Default Falloff Curve panel graph.

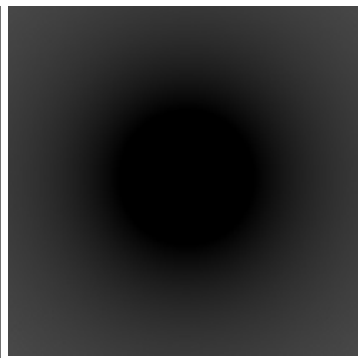


Render showing the Custom Curve lamp falloff type effect with default settings.

If you want to have a light attenuation profile that gets more intense as it moves away from the light source, you could alter the graph as below:



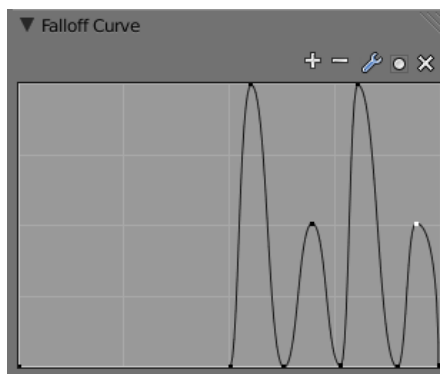
Falloff Curve for reversed attenuation.



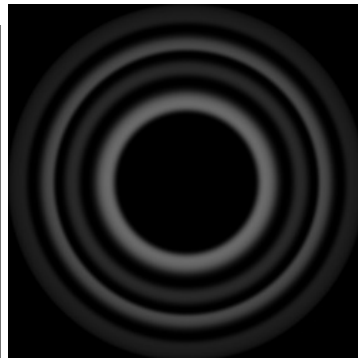
Falloff Curve for reversed attenuation rendered.

You are obviously not just limited to simple changes such as reversing the attenuation profile, you can have almost any profile you desire.

Here is another example of a different Falloff Curve profile graph, along with its resultant render output:

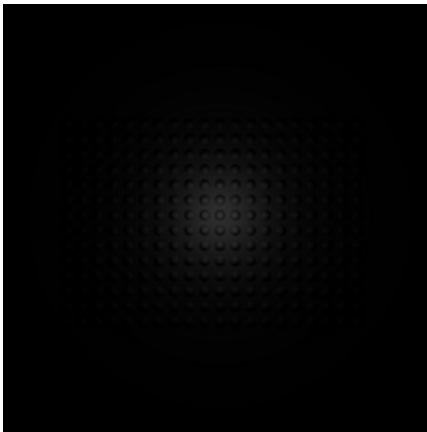


Oscillating attenuation profile.



Render showing the effects of a "wavelet" profile graph on the light attenuation.

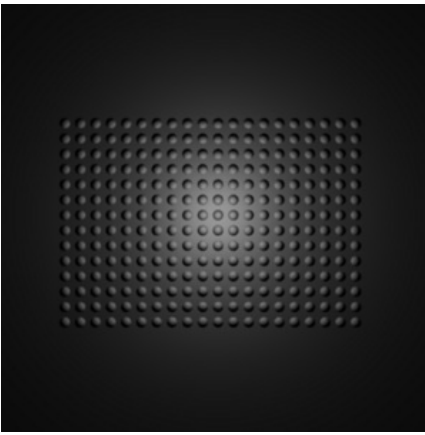
Inverse Square



Render showing the Inverse Square lamp falloff type effect with default settings.

This lamp falloff type attenuates its intensity according to inverse square law, scaled by the Distance value. Inverse square is a sharper, realistic decay, useful for lighting such as desk lamps and street lights. This is similar to the old Quad option (and consequently, to the new Lin/Quad Weighted option with Linear to **0.0** and Quad to **1.0**), with slight changes.

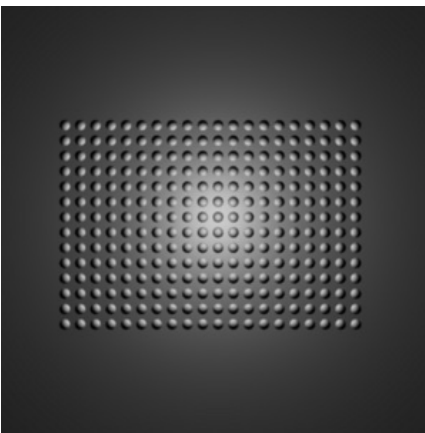
Inverse Linear



Render showing the Inverse Linear lamp falloff type effect with default settings.

This lamp falloff type attenuates its intensity linearly, scaled by the Dist value. This is the default setting, behaving the same as the default in previous Blender versions without Quad switched on, and consequently, like the new Lin/Quad Weighted option with Linear to **1.0** and Quad to **0.0**. This isn't physically accurate, but can be easier to light with.

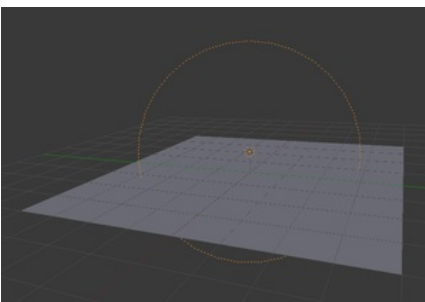
Constant



Render showing the Constant lamp falloff type effect with default settings.

This lamp falloff type does not attenuate its intensity with distance. This is useful for distant light sources like the sun or sky, which are so far away that their falloff isn't noticeable. Sun and Hemi lamps always have constant falloff.

Sphere



Screenshot of the 3D view window, showing the Sphere light clipping circle.

The Sphere option restricts the light illumination range of a Lamp or Spot lamp, so that it will completely stop illuminating an area once it reaches the number of Blender Units away from the Lamp, as specified in the Dist field.

When the Sphere option is active, a dotted sphere will appear around the light source, indicating the demarcation point at which this light intensity will be null.

The Sphere option adds a term to the chosen attenuation law, whatever it is:

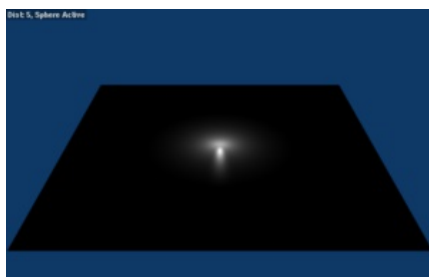
$$I' = I \times (D - r) / D \text{ if } r < D; 0 \text{ otherwise}$$

Where:

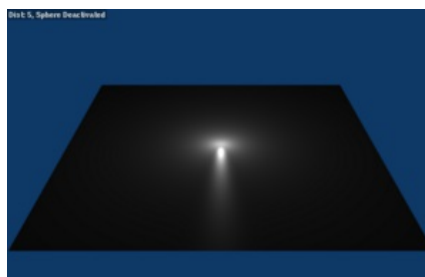
- I' is the required Intensity of light (with the Sphere option activated).
- I is the intensity of light calculated by the chosen attenuation law (without the Sphere option).
- D is the current setting of the Dist field.

- r is the distance from the lamp where the light intensity gets measured.

See the graphic at the end of the description of the Lin/Quad Weighted attenuation option.



Render showing the light attenuation of a Constant falloff light type with the Sphere option active.

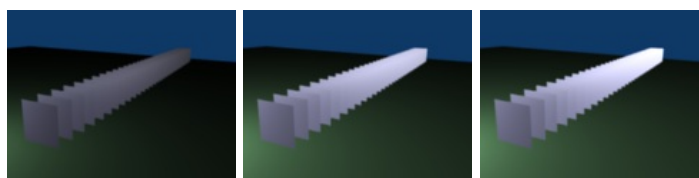


Render showing the light attenuation of a Constant falloff light type with the Sphere option deactivated.

Examples

Distance

In this example, the Lamp has been set pretty close to the group of planes. This causes the light to affect the front, middle and rear planes more dramatically. Looking at (*Various Distance settings*), you can see that as the Dist is increased, more and more objects become progressively brighter.



Distance: **10**. Distance: **100**. Distance: **1000**.
Various Distance settings (shadows disabled).

The Distance parameter is controlling where the light is falling – at a linear rate by default – to half its original value from the light's origin. As you increase or decrease this value, you are changing where this half falloff occurs. You could think of Distance as the surface of a sphere and the surface is where the light's intensity has fallen to half its strength in all directions. Note that the light's intensity continues to fall even after Distance. Distance just specifies the distance where half of the light's energy has weakened.

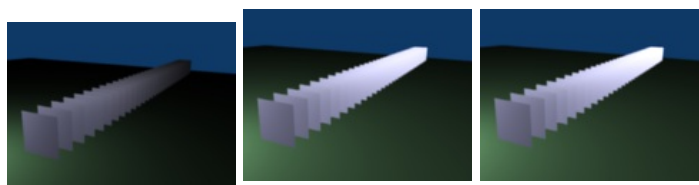
Notice in (*Distance: 1000*) that the farthest objects are very bright. This is because the falloff has been extended far into the distance, which means the light is very strong when it hits the last few objects. It is not until **1000** units that the light's intensity has fallen to half of its original intensity.

Contrast this with (*Distance: 10*), where the falloff occurs so soon that the farther objects are barely lit. The light's intensity has fallen by a half by time it even reaches the tenth object.

You may be wondering why the first few planes appear to be dimmer? This is because the surface angle between the light and the object's surface normal is getting close to oblique. That is the nature of a Lamp light object. By moving the light infinitely far away you would begin to approach the characteristics of the Sun lamp type.

Inverse Square

Inverse Square makes the light's intensity falloff with a non-linear rate, or specifically, a quadratic rate. The characteristic feature of using Inverse Square is that the light's intensity begins to fall off very slowly but then starts falling off very rapidly. We can see this in the (*Inverse Square selected*) images.



Inverse Square with **10**. Inverse Square with **100**. Inverse Square with **1000**.
Inverse Square selected (with the specified distances).

With Inverse Square selected, the Distance field specifies where the light begins to fall off faster, roughly speaking; see the light attenuation [description](#) for more info.

In (*Inverse Square with 10*), the light's intensity has fallen so quickly that the last few objects aren't even lit.

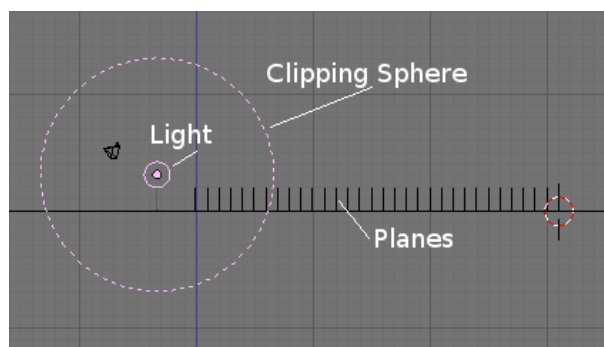
Both (*Inverse Square with 100*) and (*Inverse Square with 1000*) appear to be almost identical and that is because the Distance is set beyond the farthest object's distance which is at about **40 BU** out. Hence, all the objects get almost the full intensity of the light.

As above, the first few objects are dimmer than farther objects because they are very close to the light. Remember, the brightness of

an object's surface is also based on the angle between the surface normal of an object and the ray of light coming from the lamp.

This means there are at least two things that are controlling the surface's brightness: intensity and the angle between the light source and the surface's normal.

Sphere

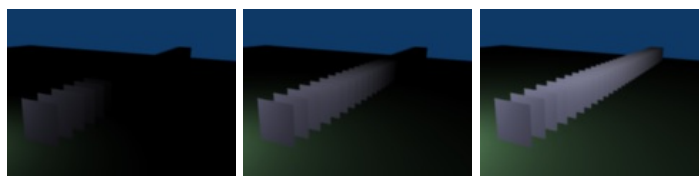


Clipping Sphere.

Sphere indicates that the light's intensity is null at the Distance distance and beyond, regardless of the chosen light's falloff. In (*Clipping Sphere*) you can see a side view example of the setup with Sphere enabled and a distance of **10**.

Any objects beyond the sphere receive no light from the lamp.

The Distance field is now specifying both where the light's rays become null, and the intensity's ratio falloff setting. Note that there is no abrupt transition at the sphere: the light attenuation is progressive (for more details, see the descriptions of the [Sphere options](#) and [light attenuations](#) above).



Sphere with **10**.

Sphere with **20**.

Sphere with **40**.

Sphere enabled with the specified distances, Inverse Linear light falloff.

In (*Sphere with 10*), the clipping sphere's radius is **10** units, which means the light's intensity is also being controlled by **10** units of distance. With a linear attenuation, the light's intensity has fallen very low even before it gets to the first object.

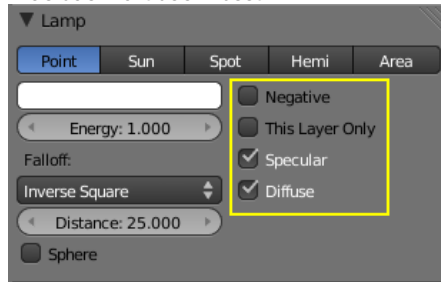
In (*Sphere with 20*), the clipping sphere's radius is now **20 BU** and some light is reaching the middle objects.

In (*Sphere with 40*), the clipping sphere's radius is now **40** units, which is beyond the last object. However, the light doesn't make it to the last few objects because the intensity has fallen to nearly **0**.

Hints

If a Lamp light is set to not cast shadows, it illuminates through walls and the like. If you want to achieve some nice effects like a fire, or a candle-lit room interior seen from outside a window, the Sphere option is a must. By carefully working on the Distance value you can make your warm firelight shed only within the room, while illuminating outside with a cool moonlight, the latter achieved with a Sun or Hemi light or both.

Was das Licht beeinflusst



Lampen Panel mit hervorgehobenen Optionen, die das Licht beeinflusst

Jede Lampe hat eine Reihe von Schaltern, die kontrollieren, welche Objekte dessen Licht erhalten und wie sie mit Materialien aufeinander wirken.

Negative

Das durch die Lampe erzeugte Licht wird auf den Oberflächen **abgezogen**, auf die es trifft. Dieses Licht macht die Oberflächen dunkel, anstatt sie zu erhellen.

This Layer Only

Bewirkt, dass die Lampe zu nur Lichtobjekten auf derselben Ebene werden.

Specular

Hält die Lampe davon ab [spiegelnde Glanzlichter](#) zu erzeugen.

Diffuse

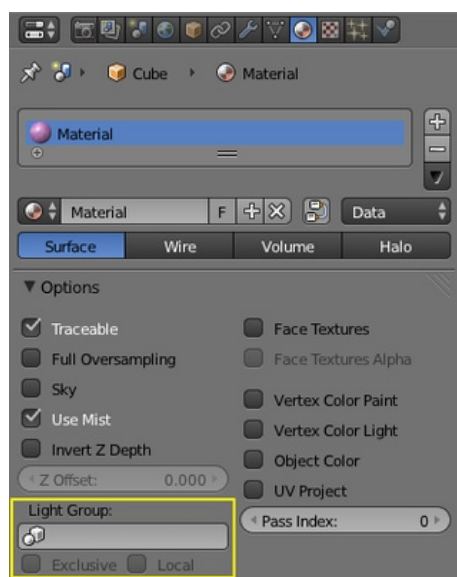
Hält die Lampe davon ab [diffuses Licht](#) zu erzeugen (es "leuchtet" Dinge nicht wirklich an).

Lampen bezogene Einstellungen

Hier sind einige Optionen, die eng mit der Lichtquelle verbunden sind, ohne Lampeneinstellungen zu sein.

Lichtgruppen

Materialien

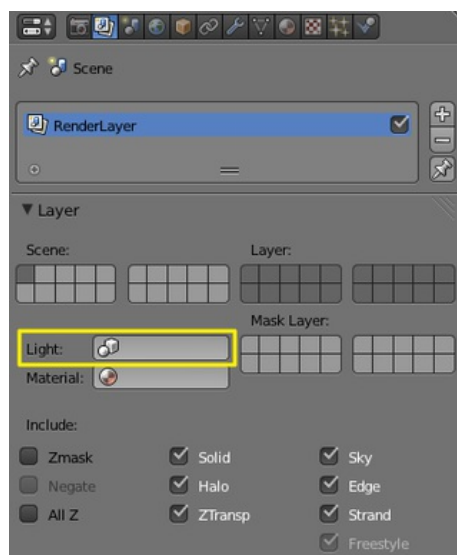


Lichtgruppen Optionen für Materialien

Standardmäßig werden Materialien durch alle Lampen in allen sichtbaren Ebenen beleuchtet, aber ein Material (und alle Objekte mit diesem Material) kann auf eine einzelne Gruppe von Lampen beschränkt werden. Diese Art der Kontrolle kann besonders in Szenen mit komplizierten Beleuchtungseinstellungen unglaublich nützlich sein. Um das zu ermöglichen, navigieren Sie zum Options Panel des Material Menüs und wählen eine Gruppe von Lampen im Light Group Feld aus. Bemerken Sie, dass eine [Lichtgruppe](#) zuerst erstellt werden muss.

Wenn die Taste Exclusive aktiviert wird, betreffen Lichter in der angegebenen Gruppe *nur* Objekte mit diesem Material.

Szene



Lichtgruppen Optionen für Render Ebenen

Es gibt ein ähnliches Steuerelement im Render Menü [Render Ebenen Panel](#). Wenn ein Lichtgruppenname im Light Feld ausgewählt wird, wird die Szene exklusiv durch Lampen in der angegebenen Gruppe beleuchtet.

Siehe auch

- [Lampen Einleitung](#)
- [Schatten](#)
- [Materialien Einleitung](#)

Volumetrisches Licht

Mit volumetrischem Licht wird das Sichtbarwerden des Lichtkegels bezeichnet. Sie können diesen Effekt immer dann beobachten, wenn sich kleine Teilchen in der Luft befinden - z.B. bei Nebel, staubiger oder verrauchter Luft.

Normalerweise sind Lichtstrahlen unsichtbar. Sie werden sichtbar, wenn sie durch Nebel, Dunst, Staub o.ä. zerstreut wurden.

Standardmäßig modelliert Blender diesen Aspekt des Lichtes nicht. Zum Beispiel, wenn Blender etwas mit einem Scheinwerfer anstrahlt, sehen Sie die Objekte und Bereiche auf dem Fußboden beleuchtet, aber nicht den Kegel/Schein des Lichtes, das aus dem Scheinwerfer kommt und unterwegs gestreut wird.

Der Schein/Kegel des Lichtes wird in der realen Welt durch Licht verursacht, das durch Partikeln in der Luft gestreut und in Ihr Auge abgelenkt wird, und dass Sie als Schein/Kegel des Lichtes wahrnehmen. Die Lichtstreuung von einer Quelle kann in Blender mit verschiedenen Optionen simuliert werden, aber ist standardmäßig nicht aktiviert.

Der einzige Lampentyp der in der Lage ist volumetrische Effekte zu erschaffen ist die [Punkt Lampe](#) (auch wenn Sie überlegen können einige der ["Himmel und Atmosphäre" Effekte](#) der Sonnen Lampen zu Nutzen).

Beispiel

[Blend Datei der Scheinwerfer Animation.](#)
[\[video link\]](#)

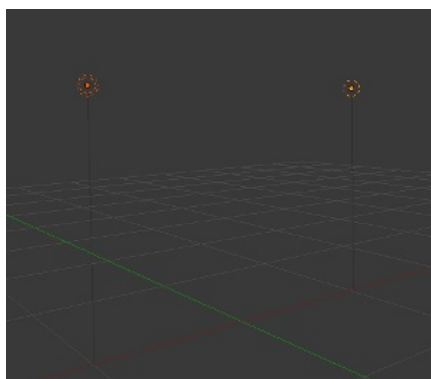
Siehe

- [Dunst](#)
- [Rauch](#)
- [Volumetrische Materialien](#)

Lampen

Blender ist mit fünf verschiedenen Lampe-Typen ausgestattet, jeder mit seinen eigenen einzigartigen Stärken und Beschränkungen. Hier sind die verfügbaren Lampen:

- [Punkt](#) ist eine Allrichtungspunkt-Licht-Quelle, die einer Glühbirne ähnlich ist.
- [Scheinwerfer](#) ist eine Richtungspunkt-Licht-Quelle, die einem Scheinwerfer ähnlich ist.
- [Bereich](#) ist eine Quelle, die Flächenproduzierendes-Licht, als Fenster, Neon, Bildschirme simuliert.
- [Halbkugel](#) simuliert eine sehr breite und weit entfernte Lichtquelle wie im Himmel.
- [Sonne](#) simuliert eine sehr weit entfernte Punktlichtquelle, wie die Sonne.



Visualisierungshöhe und Schattenmarkierung von zwei Punkt-Lampen. Strahl-Schatten wird auf der linken Lampe ermöglicht.

Sie können neue Lampen zu einer Szene mit dem Menü Hinzufügen in der oberen Leiste, oder mit (⇧ ShiftA » Hinzufügen » Lampen) hinzufügen.

Einmal hinzugefügt wird eine Position einer Lampe in der 3D-Ansicht durch einen festen Punkt in einem Kreis angezeigt, aber die meisten Typen zeigen auch gepunktete Drahtgitter, die helfen ihre Orientierung und Eigenschaften zu beschreiben. Während jeder Typ verschieden vertreten wird, gibt es einige für sie alle übliche optische Anzeiger:

Schatten

Wenn Schatten ermöglicht werden, wird ein zusätzlicher gepunkteter Kreis um den festen Kreis gezogen. Das macht es leichter schnell zu bestimmen, ob eine Lampe Schatten ermöglicht.

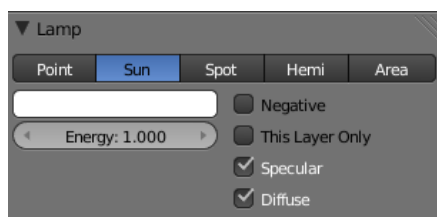
Vertikale Höhenmarkierung

Das ist eine dunkelgraue Linie, die hilft, die Position der Lampe hinsichtlich der globalen X-Y Ebene ausfindig zu machen.

Sun Lamp

Eine Sun lamp strahlt Licht konstanter Stärke in eine bestimmte Richtung ab. Eine Sun lamp kann sehr nützlich sein, wenn man eine einheitliche Tageslichtbeleuchtung im offenen Raum benötigt. In der 3D Ansicht wird die Sun lamp als Punkt mit schwarz punktierter kreisförmiger Umrandung dargestellt, von der Strahlen ausgehen, plus einer gestrichelten Linie, die die Richtung des Lichtes angibt.

Diese Richtung kann wie bei jedem anderen Objekt durch Rotieren der Sun lamp geändert werden. Da aber das Licht in eine konstante Richtung abgestrahlt wird, beeinflusst die Platzierung einer Sun lamp nicht das Render-Ergebnis (außer man benutzt die ["sky & atmosphere" Option](#)).



Sun lamp panel

Lamp options

Energy und Color

Diese Einstellungen sind für fast alle Lampen gleich und werden in [Light Properties](#) beschrieben.

Negative, This Layer Only, Specular, und Diffuse

Diese Einstellungen kontrollieren, was die Lampe beeinflusst, wie in [What Light Affects](#) beschrieben.

Die Sun lamp hat keine "light falloff" Einstellung: Sie benutzt immer eine konstante Abschwächung gleich Null (d.h. keine Abschwächung!).

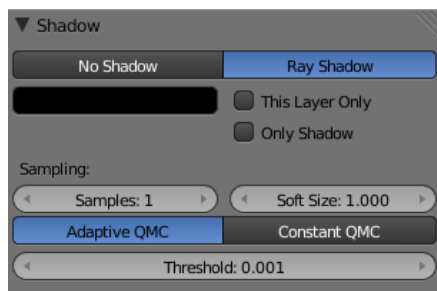
Sky & Atmosphere



Sky & Atmosphere panel

Es existieren verschiedene Einstellungen für das Aussehen der Sonne am Himmel und für die Atmosphäre, durch welche sie scheint. Für Infos dazu siehe [Sky and Atmosphere](#).

Shadow

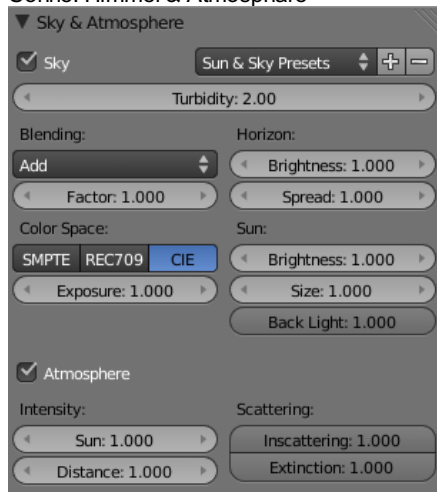


Shadow panel

Die Sun lamp kann nur ray-traced Schatten werfen. Sie besitzt die selben Schatten-Optionen wie die anderen Lampen-Typen, beschrieben in [Shadows Properties](#).

Die ray-traced Schatteneinstellungen dieser Lampe werden mit anderen Lampen geteilt, welche in [Raytraced Properties](#) beschrieben werden.

Sonne: Himmel & Atmosphäre

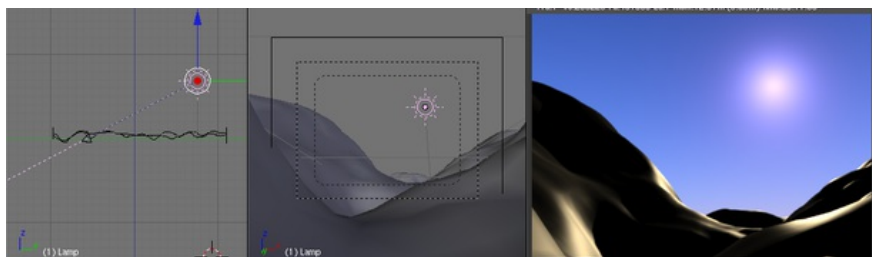


Himmel & Atmosphäre Panel

Dieses Panel erlaubt Ihnen verschiedene Eigenschaften von echtem Himmel und echter Atmosphäre zu simulieren: This panel allows you to enable a new effect that simulates various properties of real sky and atmosphere: die Streuung des weißähnlichen Sonnenlichtes während es Kilometer an Luft Overhead kreuzt. Zum Beispiel, wenn die Sonne oben steht, ist der Himmel blau (und der Horizont weißähnlich). Wenn die Sonne nahe dem Horizont ist, ist der Himmel dunkel blau/lila, und der Horizont wird orange. The dispersion of the atmosphere is also more visible, when it is a bit foggy: the more away an object is, the more “faded” in light grey it is... Go out in the countryside, a nice hot day, you will see.

To enable this effect, you have to use a Sun light source. If, as usual, the *position* of the lamp has no importance, its *rotation* is crucial: it determines which hour it is. As a start point, you should reset rotation of your Sun (with AltR, or typing **0** in each of the three Rotation fields X/Y/Z in the Transform Properties panel – N). This way, you'll have a nice mid-day sun (in the tropics).

Now, there are two important angles for the Sky/Atmosphere effect: the “incidence” angle (between light direction and X-Y plane), which determine the “hour” of the day (as you might expect, default rotation – straight down – is “mid-day”, a light pointing straight up is “mid-night”, and so on...). And the rotation around the Z axis determines the position of the sun around the camera.



The dashed “light line” of the Sun lamp crossing the camera focal point.

In fact, to have a good idea of where the sun is in your world, relative to the camera in your 3D view, you should always try to have the dashed “light line” of the lamp crossing the center of the camera (its “focal” point), as shown in (*The dashed “light line” of the Sun lamp crossing the camera focal point*). This way, in camera view (0 NumPad, center window in the example picture), you will see where will be the “virtual” sun created by this effect.

It is important to understand that the *position* of the sun has no importance for the effect: only its *orientation* is relevant. The position might just help you in your scene design.

Optionen

Sonne und Himmel Presets

- Klassisch:
- Wüste:
- Berg:

Himmel

Himmel

Dieser Knopf aktiviert die Himmel Eigenschaften: er wird einen “Himmel”, mit einer “Sonne” falls sichtbar, und mischt es mit dem Hintergrund wie in den World Einstellungen definiert wurde.

Trübheit

This is a general parameter that affects sun view, sky and atmosphere, it's an atmosphere parameters that low values describe clear sky, and high values shows more foggy sky. In general, low values give clear, deep blue sky, with “little” sun; high values give more reddish sky, with a big halo around the sun. Note that this parameter is one which can really modify the “intensity” of the sun lighting. See examples below.

Here are its specific controls:

Blending

- The first drop-down list shows you a menu of various mix methods. The one selected will be used to blend the sky and sun with the background defined in the World settings. The mixing methods are the same as described e.g. in the [Mix Compositing Node](#) page.
- Factor

Controls how much the sky and sun effect is applied to the World background.

Color space

These buttons allows you to select which color space the effect uses, with following choices:

- CIE
- REC709
- SMPTE
- Exposure

This numeric field allows you to modify the exposition of the rendered Sky and Sun (**0.0** for no correction).

Umgebungsbeleuchtung

Die Umgebungsbeleuchtung stellt ein von allen Seiten kommendes Licht zur Verfügung. Licht wird mittels der Raytrace Methode, welche ebenfalls beim "Ambient Occlusion" verwendet wird, berechnet. Der Unterschied ist, dass Umgebungsbeleuchtung den "umgebenden" Parameter der Material-Shading-Einstellungen allmählich in Betracht zieht, damit das Material den Betrag des/der umgebenden Lichts/Farbe erhält. Darüber hinaus kann die Farbquelle der Umgebung (weiß, Wolkenfarbe, Wolken-textur) und der Energieanteil des Lichts gewählt werden.

Versuchen Sie beide Einstellungen gleichzeitig zu Nutzen, um bessere Ergebnisse in der globalen Beleuchtung zu erzielen.

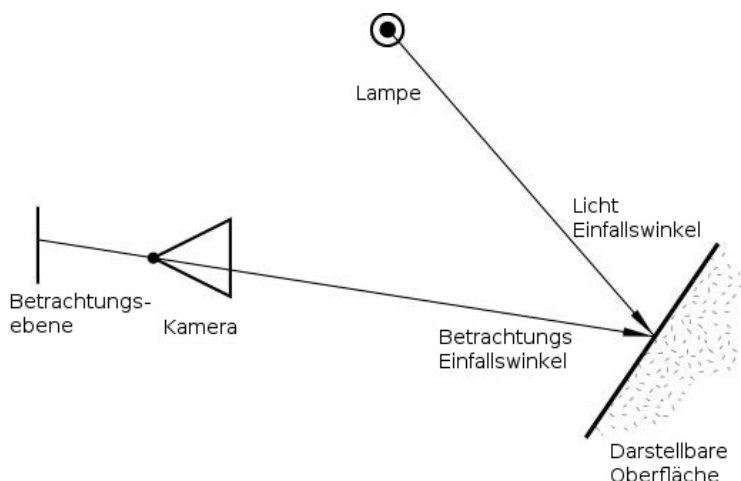
Es ist nützlich zur Nachahmung des Himmels in der Außenbeleuchtungen. Umgebungslicht kann manchmal nervend wirken.

Einleitung zu Materialien

Bevor Sie verstehen können, Materialien effektiv zum Designen zu benutzen, müssen Sie verstehen wie simuliertes Licht und Oberflächen mit Blender's Rendering Engine interagieren und wie Material-Einstellungen diese Interaktionen kontrollieren. Ein tiefes Verständnis der Engine hilft Ihnen das meiste dabei rauszuholen.

Das gerenderte Bild, dass Sie mit Blender erstellen ist eine Projektion der Szene auf einer bildlichen Oberfläche, die *Betrachtungsebene* genannt wird. Die Betrachtungsebene ist analog zu einem Film in einer traditionellen Kamera oder den Stäbchen und Zapfen im menschlichen Auge, außer dass sie simuliertes Licht empfängt, kein echtes Licht.

Zum Rendern eines Bildes einer Szene müssen wir zuerst erkennen, welches Licht aus welcher Szene an welchem Punkt in der Betrachtungsebene ankommt. Der beste Weg, dies herauszufinden besteht darin, einer geraden Linie (dem simulierten Lichtstrahl) zurück zu folgen - ausgehend vom Punkt auf der Betrachtungsebene durch den Focus-Punkt der Kamera (Standort der Kamera) bis sie auf eine *renderbare* Oberfläche innerhalb der Szene trifft. An diesem Punkt können wir nun bestimmen, welche Art von Licht diesen Punkt trifft. Die Oberflächeneigenschaften und der Lichteinfallswinkel sagen uns wie viel von dem Licht zurückreflektiert wird mittels des Einfallsbetrachtungswinkels(*Grundlegendes Prinzip der Rendering Engine.*).



Grundlegendes Prinzip der Rendering Engine.

Zwei grundlegende Phänomene treten immer und an jeder Stelle einer Oberfläche auf, wenn Licht auf sie trifft: Lichtstreuung (*diffusion*) und (mehr oder weniger) spiegelnde Reflexionen (*specular reflection*).

Diffusion and specular reflection are distinguished from each other mainly by the relationship between the incident light angle and the reflected light angle.

The shading (or coloring) of the object during render will then take into account the base color (as modified by the diffusion and specular reflection phenomenon) and the light intensity

Using the internal raytracer, other (more advanced) phenomena could occur. In raytraced reflections, the point of a surface struck by a light ray will return the color of its surrounding environment, according to the rate of reflection of the material (mixing the base color and the surrounding environment's) and the viewing angle.

On the other hand, in raytraced refractions, the point of a surface stroke by a light ray will return the color of its background environment, according to the rate of transparency (mixing the base color and the background environment's along with its optional filtering value) of the material and the optional index of refraction of the material, which will distort the viewing angle.

Of course, shading of the object hit by a light ray will be about mixing all these phenomena at once during the rendering. The appearance of the object, when rendered, depends on many inter-related settings:

- Welt (Ambiente Farbe, Radiosität, Ambiente Okklusion)
- Lichter
- Material Einstellungen (inklusive ambient, Emission, und jede andere Einstellung in jedem Panel in diesem Kontext)
- Textur(en) und ihre Kombinationen
- Materialknoten (nodes)
- Kamera
- Betrachtungswinkel
- Obstruktionen und transparente Okklusionen
- Schatten von anderen durchsichtigen/transparenten Objekten
- Render Einstellungen
- Objektdimensionen (SS settings are relevant to dimensions)
- Objektform (refractions, fresnel Effekte)

Cycles

Das Erstellen von Materialien in Cycles funktioniert mit Hilfe des Node-Editors und ist etwas anders als bei der internen Blender-Renderingengine. Trotzdem gibt es noch viele Gemeinsamkeiten.

Blender interner Render

Shader

Materialien können über zahlreiche Datenfelder (Arrays) von Eigenschaften verfügen. Es ist die Kombination all dieser Dinge, die definieren wie das Material auszusehen hat, und wie ein Shader sich verhält wenn er gerendert wurde.

Diffuse Shading

The first step is to add diffuse shading information. Diffuse shading is observed on matte and semi matte objects, or surfaces that lack highlights. It represents the microscopic variations in a surface that scatter incoming light in a very broad range of angles.

Specular Shading

In CG you can make objects appear shiny by adding specularity. Specularity represents the reflections of bright objects hitting a surface. It is important to note, that typically, specular highlights are not actually the reflections of lights. Since lights are usually much brighter than the surrounding environment, we can fake these reflections with specular highlights.

Reflektivität

To see the actual reflections of surrounding objects, shaders can be reflective, using a technique called raytracing. Raytracing casts virtual rays from the camera, and traces where they end up after bouncing off objects that have reflectivity. Not all reflections are perfectly glossy, as in a mirror or a chrome ball. Shaders can simulate non-glossy reflections, which blur the reflected image. This is expensive for render engines, since it involves a lot of computation

Transparenz

Many surfaces are transparent, or semi transparent. A shader can create transparency by simply lowering its opacity, allowing objects behind to see through, however this is rarely how transparent objects appear. Transparency can utilize raytracing as well. Most transparent objects cause refraction, which is the bending of light rays as they pass through objects of differing indexes of refraction. Just fill a glass with water to see refraction in action. Transparency can also be blurred, as in frosted glass, or some plastic containers. Blurry refractions, just like blurry reflections, are CPU intensive.

Transluzenz

If a material is able to let light through, but it scatters so much that blurred transparency is not practical, shaders can simulate translucency. For thin objects, translucency is good for things like leaves, paper, or flowers. In cg, translucency works by calculating the shading on the back side of a surface, and passing on to the other side. For thicker objects, you will want to use subsurface scattering

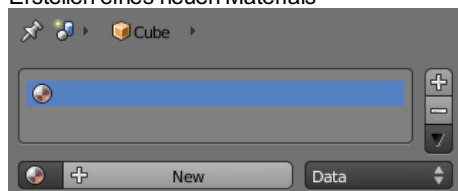
Subsurface Scattering

Subsurface scattering, or sss, is an effect that can simulate organic materials, where light is able to penetrate the surface, but bounces and scatters under the surface, and exits in a different place. This is typical in things like human skin, candles, cheese, grapes, marble, etc. SSS can add a great deal of quality and realism to materials, but it can be expensive for the renderer.


Incandescence

Some objects are, or have the appearance of being a light source, such as a light bulb or a tv screen. This can be achieved by decreasing the amount of shading information.

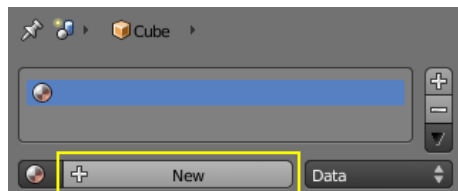
Erstellen eines neuen Materials



Material menu.

Immer wenn ein neues Objekt erstellt wird, hat es zuerst einmal kein Material zugewiesen. Durch Klicken auf das Symbol , wechseln Sie in das Shading context Menü und das Fenster für die Material-Einstellungen erscheint. Das Fenster sollte zu diesem Zeitpunkt nahezu leer sein.

Ein neues Material für das Objekt wird mit der Schaltfläche *New* erzeugt. Folgende Auswahl ist möglich:



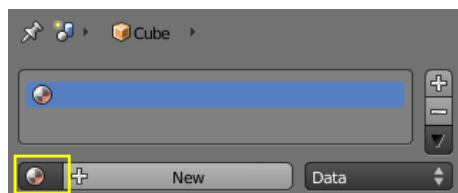
Neues Material hinzufügen.

Add New

Fügt ein neues Material in der Materialliste hinzu und verknüpft es mit dem aktiven Objekt oder *object data*. Blender wählt als Material-Name automatisch *Material.001* und numeriert selbstständig weiter.

Es ist jedoch sehr zu empfehlen, diese vorgegebenen Material-Namen durch sprechendere Bezeichnungen zu überschreiben um später einen besseren Überblick zu behalten - insbesondere wenn die Materialien mit mehreren Objekten verknüpft worden sind.

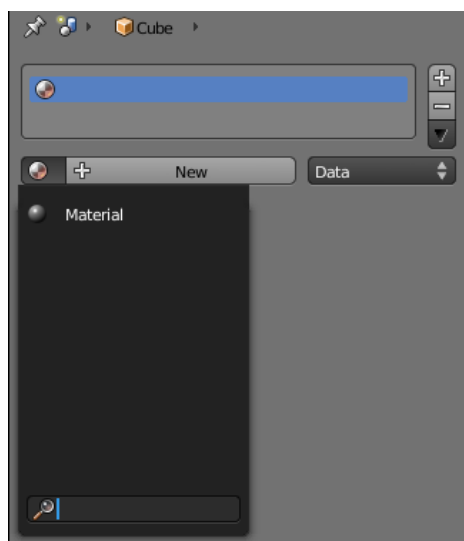
Ein Material eines anderen Objektes übernehmen



Vorhandenes Material auswählen.

Auswahl eines vorhandenen Materials

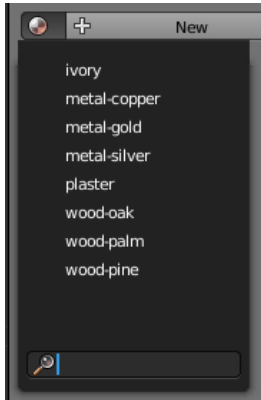
Wählen Sie ein vorhandenes Material aus der Liste aus.



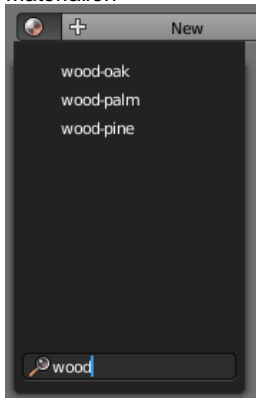
Existent material drop down menu

Blender ist so aufgebaut, dass es Ihnen möglich ist, zwischen vielen Objekten *alles* wiederzuverwenden, einschließlich Material-Einstellungen. Anstatt eine Vielzahl gleicher Materialien manuell zu erstellen, können Sie einfach bereits erstellte Materialien weiter verwenden. Hierfür gibt es zwei Möglichkeiten:


- Mit ausgewähltem Mesh klicken Sie einfach auf das kleine Kugelsymbol links vom Material-Namen. Eine Popup-Liste zeigt Ihnen alle aktuell verfügbaren Materialien an. Um eines davon zu verwenden, klicken Sie es einfach an.



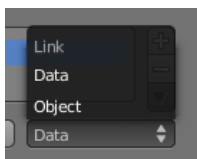
Liste der verfügbaren Materialien



Gefilterte Liste

- Neu ab Vers. 2.5 ist das *Suchen-Feld* am Ende der Material-Liste. Durch Eingabe des Materialnamens (im Beispiel *wood*) werden alle vorhandenen Materialien gefiltert und nur diese angezeigt, die den Suchbegriff beinhalten.
- In der 3D-Ansicht können Sie schnell und einfach mit dem Tastenkürzel CtrlL allen ausgewählten Objekten das Material des aktiven Objekts ([active object](#)) zuweisen. Dies ist sehr nützlich, wenn Sie einer Vielzahl von Objekten dasselbe Material zuweisen möchten. Hierzu einfach die gewünschten Objekte nacheinander auswählen ⇧ Shift RMB , zum Schluss das Objekt, das das Material bereits enthält, und CtrlL drücken. Das Material wurde nun allen ausgewählten Objekten zugewiesen.

Linking material to object or object's data



Link material to object or to object's data

A material can be linked to either the object or to the object's data.

If a material is linked to the Data, all other objects using the same data (ObData) will share the same material. But if it is linked to the object instead, the material is specific to that object only. This is helpful if you are using linked duplicates (Alt + D). This is confusing but it is an important topic, for more [click here](#).

Materials Options

Once the object has at least one material linked to it, many new panels are readily displayed to allow you to precisely control the shading of the material. Using each of these panels is discussed in the [next section](#).

The [preview panel](#) attempts to show you what the shader will produce for different kinds of geometric basic shapes. Depending on your buttons window layout, some panels may be collapsed.

Einführung in Texturen

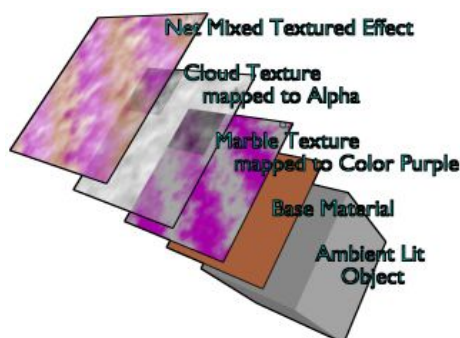
In der CGI, ist Textur-Mapping eine Methode zum Hinzufügen von Details zu Oberflächen durch das Projizieren von Bildern und Mustern auf diese Oberflächen. Die projizierten Bilder und Muster können eingestellt werden dass sie nicht nur die Farbe bestimmen, sondern auch die Spiegelung, Reflexion, Transparenz und sogar gefälschte 3-dimensionale Tiefe. Meistens werden die Bilder und Muster während dem Prozess des Renderns projiziert, aber Textur-Mapping wird auch im Skulptur-, Zeichenmodus und bei der Deformation von Objekten verwendet.

In Blender können Texturen:

- auf ein *Material* angewandt werden
- auf den [Welt Hintergrund](#) angewandt werden
- auf einen *Pinzel* angewandt werden, siehe beispielsweise:
 - ['Bildhauer' Modus](#)
 - [Zeichnen der Textur](#)
- assoziiert mit Modifizierern, siehe:
 - Partikel Texturen
 - Ozean Texturen

Texturen für Materialien

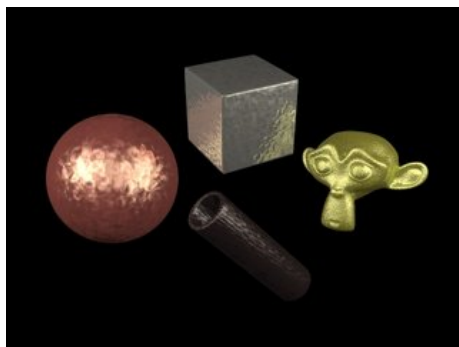
The material settings that we've seen so far produce smooth, *uniform* objects, but such objects aren't particularly true to reality, where uniformity tends to be uncommon and out of place. In order to deal with this unrealistic uniformity, Blender allows the user to apply *textures* which can modify the reflectivity, specularity, roughness and other surface qualities of a material.



Textures Layer on base Material

Texturen sind wie zusätzliche Schichten die auf dem Basismaterial liegen. Texturen beeinflussen ein oder mehrere Aspekte der Netto-Färbung des Objektes. Die Netto-Färbung die Sie sehen ist eine Sorte von Schichten aus Effekten, die in diesem Beispielbild gezeigt wird. Die Schichten, wenn Sie möchten, sind:

1. Ihr Objekt leuchtet mit **Umgebungs** Licht based on your world settings.
2. Ihr Basis **Material** färbt die ganze Oberfläche in einer einheitlichen Farbe, die auf Licht reagiert, giving different shades of the diffuse, specular, and mirror colors based on the way light passes through and into the surface of the object
3. We have a **primary texture** layer that overlays a purple marble coloring.
4. We next have a **second cloud texture** that makes the surface transparent in a misty/foggy sort of way by affecting the Alpha value
5. These two textures are **mixed** with the base material to provide the net effect; a cube of purplish-brown fog.



Some Metal Textures

This notion of using *more than one* texture, to achieve a combined effect, is one of the "hidden secrets" of creating realistic-looking objects. If you carefully "look at the light" while examining any real-life object, you will observe that the final appearance of that object is best described as the combination, in different ways and in different amounts, of several distinct underlying visual characteristics. These characteristics might be more (or less) strongly apparent at different angles, under different lighting conditions, and so forth. Blender allows you to achieve this in many ways. You can use "a stack of texture layers" as described in [this section](#), or you can also use arbitrarily-complex networks ("noodles"...) of "texture nodes" as discussed [here](#), the choice is yours.

Texturen für Materialien fallen in drei primäre Kategorien:

Prozedurale Texturen

Texturen die mittels einer mathematischen Formel generiert worden sind. Zum Beispiel, Holz, Wolken, und verzerrtes Geräusch

Bilder und Filme

Fotos und Filme die auf Objekte projiziert werden. Zum Beispiel, eine flache Karte der Erde dargestellt auf einer Kugel.

Umgebungskarten

Texturen die zum Erstellen von Impressionen von Reflexionen und Refraktionen genutzt werden. Zum Beispiel, ein Bild einer Straße reflektiert in einem Autofenster.

Daten oder Modifizierer Texturen

Texturen die aus Rohdaten oder einem bestimmten Modifizierer in der Szene gewonnen werden.
Zum Beispiel:

- volumetrisches Material das Voxel Daten Texturen oder punktdichte Texturen nutzt
- Texturen die aus einem Ozean Modifizierer gewonnen werden können

note for editors

Hier unten sollte ein einzuleitender Text für alle Typen von Texturen gezeigt werden

Welt Texturen

Todo

Pinsel Texturen

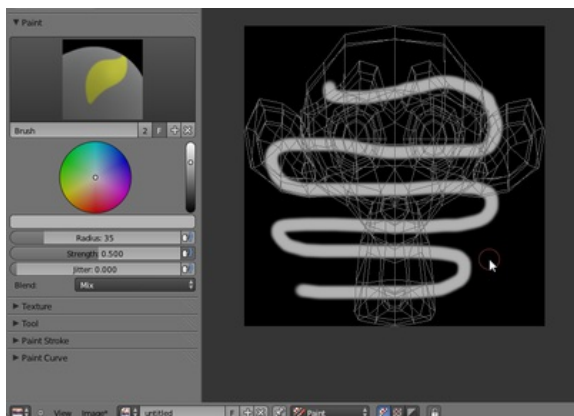
Todo

Texturen malen (Texture Painting)

Eine UV-Textur ist ein Bild (Foto/Zeichnung/Malerei, Sequenz oder Film), um die Oberfläche eines Meshes farbig zu gestalten. Die UV-Textur wird auf das Mesh mittels einer oder mehrerer UV Maps gemappt. Es gibt drei Verfahren, dieses Bild, auf dem die UV-Textur basiert, aufbringen zu können:

- Malen eines zweidimensionalen Bildes im UV/Image Editor auf die aktuell ausgewählte UV-Textur, wobei dessen UV-Map benutzt wird, um die Farben den Mesh-Faces korrekt zuzuordnen.
- Bemalen des Meshes in der 3D-Ansicht, wobei man Blender die aktuell ausgewählte UV Map benutzt, um die UV-Textur entsprechend zu aktualisieren (Siehe auch "[Projection Painting](#)").
- Benutzen eines beliebigen Bildbearbeitungs-(Mal-)Programms, um ein Bild zu erzeugen. Hierbei wird im UV/Image Editor die UV-Textur ausgewählt und dann dieses Bild geladen. Blender wird daraufhin die UV Map dieser Textur nutzen, um die Farben auf den Faces des Meshes zu applizieren.

Blender bietet eine eingebaute Malfunktion namens Texture Paint, die speziell dabei helfen soll, die UV-Texturen und Bilder schnell und leicht entweder im UV/Image Editor oder in der 3D-Ansicht zu bearbeiten. Weil eine UV-Textur lediglich ein Bild für einen besonderen Zweck darstellt, kann man im Prinzip auch jedes andere externe Malprogramm hierfür verwenden. Beispielsweise ist Gimp ein mit allen notwendigen Funktionen ausgestattetes Bildbearbeitungsprogramm, das ebenfalls **open source** ist.



Texturen malen in Blender

Weil ein Mesh mehrere Ebenen (Layer) von UV-Texturen haben kann, können auch mehrere Ebenen vorhanden sein, die dem Mesh seine Farbigkeit geben. Gleichwohl besteht jede UV-Textur immer nur aus genau einem Bild.

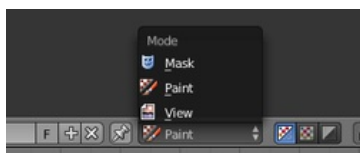
Texture Paint funktioniert sowohl im 3D-Ansichtsfenster als auch im UV/Image Editor. In der dreidimensionalen Ansicht kann man im Texture Paint Mode direkt auf das Mesh malen durch [Projektion auf die UVs](#).

Erste Schritte

Sobald man das flächige Projizieren (Unwrap) des Modells in eine UV-Textur durchgeführt hat (siehe vorangegangene Seiten), kann man mit dem Texturierungsprozess beginnen. Man kann im Texture Paint Mode nicht direkt auf ein Mesh malen, ohne **zuvor** ein Unwrap für dieses Mesh ausgeführt zu haben **und** einen der folgenden Schritte vollzogen zu haben. Die Alternativen hierbei sind:

- [Load an image](#) Laden eines schon vorhandenen Bildes in den UV/Image Editor mittels Image->Open->select file
- [Create a new image](#) Erstellen eines neuen Bildes (Image->New->specify size).

Hat man diese beiden Punkte abgehakt, kann man das Bild mit dem Texture Paint Mode bearbeiten, sobald man alternativ dies getan hat:



Texture Paint Mode aktivieren

- Nach dem Auswählen der Option "Texture Paint Mode" aus der Modusauswahl auf dem Header unterhalb des 3D-Ansichtsfensters kann man sogleich direkt auf das Mesh malen.
- Im Bearbeitungskontext des UV/Image Editors stellt man von "View" auf "Paint" um (siehe rechts).

Vielfache von 2

Texture Paint arbeitet sehr flüssig und spricht gut an, wenn man sich im 3D-Ansichtsfenster befindet und wenn das Bild ein Quadrat mit Maßen ist, die ein Vielfaches von 2 sind; 256x256, 512x512, 1024x1024 usw.

Ist Texture Painting einmal aktiviert, wird die Maus zum Pinsel. Um wieder mit der UV Map zu arbeiten (zum Beispiel wenn man Koordinaten verschieben möchte), muss man auf "View" zurückstellen.

Sobald man Texture Painting aktiviert hat oder in den Texture Paint Mode gewechselt ist, werden Pinseleinstellung im Properties Panel (N) dargestellt.

Im UV/Image Editor malt man hingegen auf eine flache Leinwand, die mittels der UV-Koordinaten um das Mesh gelegt wird. Jede Änderung hier im UV/Image Editor erscheint sofort auch in der 3D-Ansicht und umgekehrt.

Eine ganze Reihe weiterer Pinsel und Farben kann aus dem Properties Panel im UV/Image Editor gewählt werden. Erfolgte Pinselwechsel wirken sich jeweils immer auch gleich auf das andere Panel aus. Allerdings sollte man beachten, dass die bearbeitete Textur **nicht** automatisch gespeichert wird; dies muss ausdrücklich von Hand erledigt werden im UV/Image Editor über Image->Save.

Texturvorschau

Wird eine Textur bereits verwendet, um dem Modell in der Scene Farbe zu verleihen, diesem als Bumpmap oder als Displacement Map zu dienen oder ihm Alpha-Transparenz hinzuzufügen (technisch ausgedrückt: wenn sie schon für einen bestimmten Zweck über den Texturkanal auf Grundlage eines UV-Eingangskanals gemappt worden ist), kann man die Auswirkungen seiner Malereien in der Scene im Moment des Malens direkt mitverfolgen.

Zu diesem Zweck erstellt man nebeneinander zwei Fenster, wobei das 3D-Ansichtsfenster in den Textured-Anzeigemodus versetzt wird, während das zweite Fenster als UV/Image Editor eingerichtet und das zu bearbeitende Bild dort hinein geladen wird. Man stellt die 3D-Ansicht so ein, dass man das Objekt sehen kann, für das zum geladenen Bild eine UV-Map vorliegt. Ein Vorschaufenster wird geöffnet und über dem Objekt angeordnet (für mehr Informationen siehe 3D-Ansichtsoptionen). Im Bild rechts ist das gemalte Bild auf "Normal" gemappt, was man als "Bumpmapping" bezeichnet, wobei die Graustufen einer planen Oberfläche eine unregelmäßige Struktur verleihen. Für weitere Informationen über Bumpmapping siehe unter Texture Mapping Output.

Pinseleinstellungen

Um das Properties Panel im UV/Image Editor aufzurufen, drückt man N. In diesem Panel kann man eine Vielzahl von Pinseln (Brushes) erstellen, wobei man jeweils individuelle Einstellungen (z.B. für Farbe und Stärke) vornehmen kann. Man benutzt die Pinselauswahl (Brush Selector), um die Pinsel zu wechseln oder einen neuen zu erstellen. Fügt man einen Pinsel hinzu, ist der neue erst einmal eine genaue Kopie des Originals. Man kann aber sogleich die Einstellungen für den neuen Pinsel einstellen. Texture Paint kann eine unbegrenzte Zahl von Pinseln und benutzerdefinierten Einstellungen für diese aufnehmen, die sich im Paint-Panel festlegen lassen.

Um einen Pinsel zu benutzen, klickt man auf dessen Namen. Befinden sich mehr Pinsel in der Auswahl, als gleichzeitig angezeigt werden können, sind diese unter Benutzung der Auf- und Abwärtspfeile zu erreichen. Man benennt einen neuen Pinsel durch Klicken in das Namensfeld und Eingabe eines sinnfälligen Namens nach Wunsch, z.B. "Red Air" für einen roten Airbrush. Zum Entfernen eines Pinsels klickt man auf diesen und drückt auf die Taste X neben dessen Namen. Möchte man hingegen den Pinsel beim nächsten Öffnen von Blender doch wieder zur Verfügung haben, betätigt man die Taste F gleich neben der Pinsel-Lösch taste mit dem X.

Besitzt man ein drucksensitives Grafiktablett, kann man die kleine Taste "P" drücken, die sich in der Nähe der Einstelltasten für Opacity (Deckkraft), Size (Größe), Falloff (Einflussbereich) und Spacing (Zwischenräume) befindet, welche den Einfluss des Stifts in Abhängigkeit vom ausgeübten Druck steuern. Benutzt man die Radierfunktion des Stifts, wird in den Erase Alpha Mode umgeschaltet.

Man kann S an einem beliebigen Punkt des Bildes klicken, um wie mit einer Pipette die dortige Farbe als neue Pinselfarbe aufzunehmen.

Paint (Malen)



Pinseleinstellungen

Brush presets

Auswahl einer Pinseleinstellung. Die meisten Pinsel haben dieselben Einstellungen.

Enable Pressure Sensitivity

Das Icon rechts von den drei folgenden Einstellungen schaltet die Drucksensitivität des Tablets ein und aus, mit dem die Stärke eines Effekts gesteuert werden kann.

Color

Pinselfarbe.

Radius

Der Radius f des Pinsels in Pixeln angegeben.

Strength

Die Kraft, mit welcher der Pinsel angewendet wird.

Jitter

Definiert, wie zitterig-unruhig der Farbauftrag im Umkreis der Pinselposition erfolgen soll.

Blend

Legt fest, in welcher Weise die Farbe auf einer darunterliegenden Textur aufgetragen wird.

- Mix: Die Pinselfarbe wird zu bereits vorhandenen Farben hinzugemischt.
- Add: Die Pinselfarbe wird einer vorhandenen Farbe hinzugefügt; grün zu rot ergibt gelb.
- Subtract: Die Pinselfarbe wird abgezogen; malt man mit blau auf lila, entsteht rot.
- Multiply: Der RGB-Wert der Grundfarbe wird mit der Pinselfarbe multipliziert.
- Lighten: Der RGB-Wert der Grundfarbe wird durch die Pinselfarbe aufgehellt.
- Darken: Abdunkeln der Farben.
- Erase Alpha: Macht das Bild an den Stellen, an denen gemalt wird, transparent, so dass Hintergrundfarben sowie Farben auf darunterliegenden Ebenen durchscheinen können. Wenn man 'malt', wird der falsche Schachbrettmuster-Hintergrund verborgen.
- Add Alpha: Macht das Bild durch Farbauftrag deckener.

Um die Ergebnisse der Erase Alpha- und Add Alpha-Mischmodi im UV/Image Editor sehen zu können, muss man die Alphakanal-Anzeige durch Klicken auf die Tasten "Display Alpha" oder "Alpha-Only" einschalten. Transparente Bereiche (nicht Alpha) werden mit einem Schachbrettmuster-Hintergrund dargestellt.

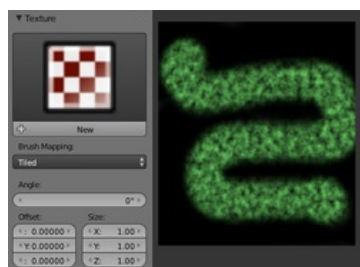
Image

Wenn man den Clone-Pinsel benutzt, kann man ein Bild als Clone-Quelle auswählen.

Alpha

Deckkraft der Clone-Bilddarstellung.

Texture (Textur)



Textur-Optionen und Beispiel

Man benutzt die Texturauswahl unter dem Paint-Panel, um eine zuvor geladene oder prozedurale Textur als Pinselmuster zu verwenden. Hierbei ist zu beachten, dass man vor der Benutzung ein Platzhalter-Material definiert haben muss und dass die jeweils ausgewählte Textur auch mittels der Material- und Texture-Tasten definiert worden ist. Es ist indessen nicht erforderlich, dass dieses Material oder die Textur auf jeden Meshbereich überall zugeordnet werden muss; es muss einfach nur definiert sein. Das Beispiel rechts zeigt die Auswirkungen des Malens mit einer flachen (gemaserten) Holztextur. Stellt man diese Textur auf Ringe um, erhält man ein Pinselmuster, das an eine Zielscheibe oder eine Blume erinnert.

Bitte beachten: Im Clone-Malmodus wechselt dieses Feld in eine Anzeige des Bildes oder der Textur, das/die als Vorlage für das Clonen benutzt wird.

Brush Mapping

Festlegen, wie die Textur auf den Pinsel angewendet wird

View Plane

Beim zweidimensionalen Malen bewegt sich die Textur mit dem Pinsel.

Tiled

Textur wird durch die Pinselposition versetzt.

3d

Entspricht dem Tiled-Mapping.

Offset

Versatz der Textur in x, y und z.

Size

Achsenbezogenes Festlegen der Texturskalierung.

Tool (Werkzeug)

Es gibt vier verschiedene Pinselarten:

Draw

Der normale Pinsel malt einen Streifen Farbe.

Soften

Schafft sanfte Übergänge zwischen zwei Farben.

Smear

Wenn man klickt, werden die Farben unter dem Mauszeiger erfasst und in Bewegungsrichtung der Maus vermengt. Ähnelt dem Wischfinger in *Gimp*.

Clone

Kopiert die Farben eines ausgewählten Bildes (hier als Beispiel Tex.Dirt) in das gerade bearbeitete Bild. Das Hintergrundbild wird angezeigt, wenn dieser Pinsel ausgewählt ist; man benutzt den Blend-Regler, um die Intensität des Hintergrundbildes einzustellen.

Paint Stroke (Pinselstrich)**Airbrush**

Die Funktion wird angewendet, solange die Maustaste gehalten wird.

Rate

Abstände zwischen den Farben beim Airbrushen.

Space

Begrenzen des Pinselauftrags auf einen Distanzwert.

Distance

Abstand zwischen Pinseltupfern, angegeben als Prozentwert des Pinseldurchmessers.

Wrap

Setzt den Farbauftrag auch auf den gegenüberliegenden Bildseite fort, sobald man mit dem Pinsel über die Kante der **anderen** Seite drüberwegmalt (für jede Seite, oben/unten, links/rechts). Sehr nützlich für das Erstellen nahtloser Texturen.

Paint Curve (Malkurve)

Die Malkurve erlaubt die Steuerung des Einflussbereichs des Pinsels. Durch Verändern der Kurve kann man den Pinsel weicher oder härter machen.

Speichern

Wenn im Header-Menü ein Sternchen neben dem Menü "Image" erscheint, bedeutet dies, dass Änderungen am Bild vorgenommen worden sind, die noch nicht abgespeichert worden sind. Man benutzt die Option Image->Save As Image in demselben Menü, um das Bild entweder mit einem anderen Namen zu speichern oder ein bestehendes zu überschreiben.

UV-Layouts

Da Bilder, die als UV-Layouts dienen, sich in ihrer Funktion von anderen Bildern unterscheiden, sollten sie in einem gesonderten Verzeichnis abgelegt werden.

Das Bilddatei-Format für das Abspeichern ist unabhängig vom Render-Format. Das Format, um im "Save As Image"-Fenster ein UV-Layout zu speichern, ist standardmäßig PNG (.png).

Sofern entweder "Packing" im Fensterkopf aktiviert ist oder man manuell Image->Pack Image auswählt, ist ein separates Speichern der Bilder in eigene Dateien nicht erforderlich.

Benutzen eines externen Bildbearbeitungsprogramms

Möchte man ein externes Bildbearbeitungsprogramm benutzen, um die UV-Textur zu bearbeiten, muss man wie folgt vorgehen:

1. Starten des Programms (Gimp, Photoshop etc.),
2. Öffnen des Bildes oder Anlegen eines neuen Bildes,
3. Bearbeiten des Bildes und
4. Abspeichern innerhalb des jeweiligen Programms.

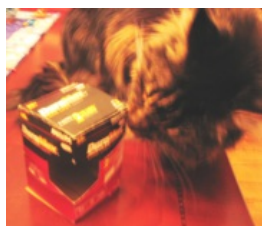
5. Zurück in Blender lädt man das Bild in den UV/Image Editor.

Man sollte dann ein externes Programm nutzen, wenn Gruppen von Leuten mit verschiedenen Programmen an der Entwicklung der UV-Texturen arbeiten, wenn man bestimmte Spezialeffekte anwenden möchte, die Texture Paint nicht bietet, oder wenn man einfach mit dem Malprogramm seiner Wahl vertrauter ist.

UV Mapping

Der am Meisten flexible Weg eine 2D Textur auf ein 3D Objekt anzuwenden ist ein Prozess der "UV Mapping" genannt wird. Hier werden dreidimensionale (X, Y & Z) Meshes genommen und auf ein zweidimensionales (X & Y ... oder eher, wie wir später sehen werden, "U & V") entpackt. Farben im Bild werden somit Ihrem Mesh zugeordnet, und tauchen als Farben der Flächen im Mesh auf. Benutzen Sie UV Texturen um Realismus zu Ihren Objekten hinzuzufügen der durch prozedurale Materialien und Texturen nicht erreicht werden kann, und bessere Details bietet als das Malen mit Vertexes.

Was sind UV's ?



Verpackung wird inspiziert



Flach gelegte Verpackung

Die beste Analogie zum Verständnis von UV Mapping ist das Aufschneiden einer Verpackung. Die Verpackung ist ein dreidimensionales (3D) Objekt, genau so wie der Mesh Würfel den Sie zu Ihrer Szene hinzufügen.

Wenn Sie ein Schere nehmen würden und eine Naht oder einen Schlitz in die Verpackung schneiden würden, würden Sie in der Lage sein sie flach auf einen Schreibtisch zu legen. Während Sie auf die Verpackung auf dem Tisch sehen, könnten wir sagen das U die Richtung von links nach rechts darstellt, und V die Richtung von oben nach unten. Dieses Bild hat also zwei Dimensionen (2D). Wir benutzen **U** und **V** um auf die "Texturraum-Koordinaten" zu verweisen, anstatt **X** und **Y**, welche immer genutzt werden (entlang der **Z** Achse) um auf den "3D-Raum" zu verweisen.

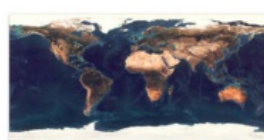
Wenn die Verpackung wiederhergestellt ist, wird eine bestimmte UV Position auf dem Papier an eine (X,Y,Z) Position der Verpackung transferiert. Dass ist was der Rechner mit einem 2D Bild tut wenn er es auf ein 3D Objekt anwendet.

Während des Auspackens des UV, sagen Sie Blender genau wie die Flächen Ihres Objektes gemappt werden sollen (in diesem Fall, eine Verpackung) auf ein flaches Bild im UV/Bildeditor Fenster. Ihnen bleibt die Freiheit wie sie dies anstellen möchten.

(Weiterführend von unserem vorherigen Beispiel, stellen Sie sich vor, dass wenn Sie die Verpackung zuerst auf den Schreibtisch gelegt haben, Sie ihn jetzt in kleine Teile schneiden, und diese Teile strecken und/oder schrumpfen lassen, und dann diese auf eine Weise auf dem Photograph anordnen der auch auf dem Schreibtisch liegt ...)

Kartographie Beispiel

Kartographen (Ersteller von Karten) haben über ein Millennium an diesem Problem gearbeitet. Ein Kartographie (Kartenerstellungs) Beispiel erstellt eine Projektionskarte der gesamten Welt. In der Kartographie, nehmen wir die Oberfläche der Erde (als Kugel) und machen eine flache Karte die zusammengefasst in ein Handschuhfach an Board des Space Shuttles gelegt werden kann. Wir 'füllen' den Platz entlang der Pole, oder spaces toward the poles, oder ändern den Umriss der Karte auf eine der folgenden Wege:



Mercator-Projektion



Mollweide-Projektion

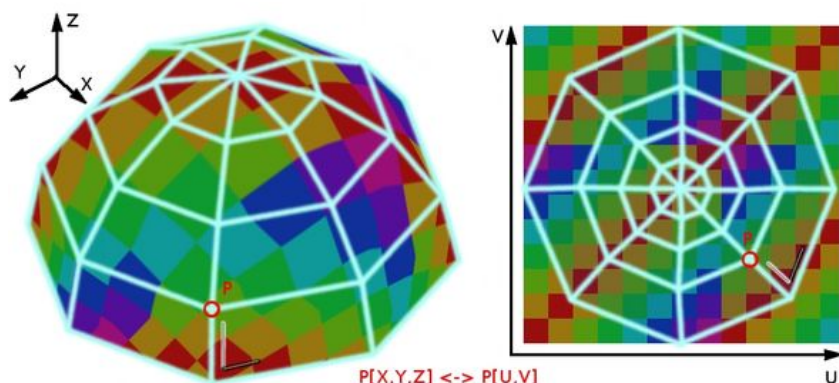


Albers-Kegelprojektion

Jede dieser Methoden ist ein Beispiel wie eine UV Karte auf eine Kugel angewendet werden kann. Jede der hundertten oder mehr bekannt, akzeptierten Sorten der Projektionen haben ihre Vor-, und Nachteile. Blender erlaubt uns das Selbe auf die von uns gewollte Weise auf den Rechner zu übertragen.

Bei komplexeren Modellen (wie auf der Welt Karte oben) kommt ein Problem auf wo die Flächen nicht "geschnitten" werden können, und anstelle gestreckt werden um sie flach zu machen. Das hilft dabei einfachere UV Karten zu erstellen, aber manchmal Verzerrungen zur endgültigen Textur hinzufügt. (Länder und Staaten die näher am Nordpol oder Südpol sind sehen auf einer flachen Karte kleiner aus als andere die sich näher am Äquator befinden.)

Halb-Kugel Beispiel



3D Raum (XYZ) gegen UV Raum (klicken zum Vergrößern)

Auf diesem Bild können sie einfach sehen dass die Form und Größe der markierten Fläche im 3D Raum anders als im UV Raum ist.

Der Unterschied entsteht durch das "Ziehen" (technisch mapping genannt) des 3D Teiles (XYZ) auf eine Fläche (z.B die UV Karte).

Wenn ein 3D Objekt eine UV Karte hat, wird jeder Punkt auf dem Objekt zusätzlich zu den 3D-Koordinaten X, Y, und Z, entsprechende U und V Koordinaten haben. (P im Bild oben ist ein Beispiel wie ein Punkt auf einem 3D Objekt auf ein 2D Bild gemappt wird.)

Der UV Editor

Der UV/Bild Editor ist der Ort wo sie UV's bearbeiten. Hier ist eine Übersicht der Werkzeuge die dort gefunden werden können. Die Benutzung des UV Editors wird in der nächsten Abteilung erklärt.

Kopfzeile

Die Kopfzeile enthält verschiedene Menüs und Option um mit UVs zu arbeiten.

Ansicht

Werkzeuge für das [Navigieren](#), arbeiten mit dem Editor und das Kontrollieren wie Dinge dargestellt werden. Das Eigenschaften Panel hat Ansichtsoptionen und Manipulationswerkzeuge. Wenn ein Bild benutzt wird, werden die Bildeigenschaften dargestellt. Das Scopes Panel wird genutzt wenn mit Bildern gearbeitet wird. Es enthält verschiedene Visualisierungen zu Bildern

Auswählen

Werkzeuge zum [Auswählen von UV's](#).

Bild Menü

Dieses hält Optionen bereit wenn Sie mit [Bildern arbeiten](#) und [Texturen zeichnen](#).

UV Menü

Dieses hält Werkzeuge für das [Meshes entpacken](#) und [Bearbeiten von UV's](#) bereit.

Bildauswahl Menü

Wählen Sie welches Bild übernommen werden soll wenn Sie mit [Mit Bildern arbeiten](#).

Pin Image

Zeigt momentanes Bild unabhängig vom ausgewählten Objekt an.

Pivot Punkt Selektor

Ähnlich zu dem Arbeiten mit Pivot Punkten in der 3D Ansicht.

Sync Auswahl

Keeps UV and Mesh component selections in sync.

Auswahl Modi

- Vertex
- Kanten
- Flächen
- Island

Sticky Selection Mode

Wenn die Sync Auswahl deaktiviert ist, kontrollieren diese Optionen wie UV's ausgewählt werden.

Proportionales Bearbeiten

Funktioniert wie [Proportionales Bearbeiten in der 3D Ansicht](#)

UV Einrastung

Ähnlich zum Einrasten in der 3D Ansicht

Aktiver UV Textur Selektor

Auswählen welche UV Textur benutzt werden soll

Eigenschaften Panel

Wachsstift

Ähnlich zum [Wachsstift](#) in der 3D Ansicht.

UV Vertex

[Transformations Eigenschaften](#) zum Auswählen von UV's

Bild

Enthält die Eigenschaft vom [Bild](#)

Ansicht

Steuert die [Anzeigeoptionen für UV's](#), und ähnliche Einstellungen wenn mit es um das [Arbeiten mit Bildern](#) geht.

Navigieren im UV Raum

Schwenken kann durch das Klicken von MMB  und ziehen erreicht werden.

Sie können durch das hoch und runter Scrollen von MMB  vergrößern und verkleinern. Sie können wenn Sie in der 3D Ansicht sind, + NumPad oder - NumPad zum Vergrößern und Verkleinern benutzen.

Die folgenden Tastenkombinationen sind verfügbar, und durch das Menü Ansicht:

- 1:8 vergrößern 8 NumPad
- 1:4 vergrößern 4 NumPad
- 1:2 vergrößern 2 NumPad
- 1:1 vergrößern 1 NumPad
- 2:1 vergrößern ⇧ Shift2 NumPad
- 4:1 vergrößern ⇧ Shift4 NumPad
- 8:1 vergrößern ⇧ Shift8 NumPad
- Alles zeigen ↵ Home
- View Center . NumPad

Ansichtsoptionen

Sie können einstellen wie UV's angezeigt werden in dem Panel Ansicht:

Outline/Dash/Black/White

Stellt ein wie UV Kanten angezeigt werden

Weich

Lässt Kanten mit Antialiasing erscheinen

Modified

Zeigt Ergebnisse des Modifikators in der UV Ansicht an

Stretch

Zeigt wie groß der Unterschied zwischen den UV Koordinaten und den 3D Koordinaten beträgt. Blau steht für eine kleine Verzerrung, Rot für eine große Verzerrung. Wählen Sie zur Anzeige der Verzerrung der Winkel oder des Bereiches.

Vorteile von UV's

Während prozedurale Texturen (in den vorherigen Kapiteln beschrieben) nützlich sind-wiederholen sie sich nie und "passen" sich immer 3D Objekten an - sie sind nicht effektiv genug für komplexe und natürliche Objekte. Zum Beispiel, wird die Haut auf einem menschlichem Kopf niemals wirklich gut aussehen wenn sie prozedural generiert wird. Falten auf einem menschlichem Kopf, oder Kratzer auf einem Auto treten nicht bei zufälligen Stellen auf, aber hängen von der Form des Models und der Nutzung ab. Manuell-gezeichnete Bilder, oder Bilder aus der realen Welt bieten mehr Kontrolle und Realismus. Für Details auf Buchcovern, Teppichen, Flecken und detaillierten Requisiten, sind Künstler in der Lage, jedes Pixel der Oberfläche über eine UV-Textur zu steuern.

Eine UV Karte beschreibt welches Teil einer Textur an welches Polygon im Model gehängt werden soll. Jedes Polygon Vertex wird 2D Koordinaten zugewiesen die definierten welcher Teil des Bildes zur Karte werden soll. Diese 2D Koordinaten werden UV's genannt (vergleichen Sie dass zu den XYZ Koordinaten in 3D). Die Operation zum Generieren von UV Karten wird "entpacken" genannt, denn es ist, als ob das Mesh auf eine 2D Fläche entfaltet wird.

Für meist simple 3D Modelle, stellt Blender ein automatisches Set an Algorithmen zum Entpacken bereit das leicht anwendbar ist. Für komplexere 3D Modelle, wie kubische, zylindrische und kugelförmige sind diese jedoch nicht geeignet. Für gleichmäßige und genaue Projektionen, benutzen Sie Nähte um das UV Mapping zu führen. Diese können beim Anwenden von Texturen auf willkürliche und komplexe Formen, wie menschliche Köpfe und Tiere helfen. Oft sind diese Texturen gezeichnete Bilder, die in Anwendungen wie Gimp, Photoshop, oder Ihrer Lieblings Anwendung zum Malen erstellt werden.

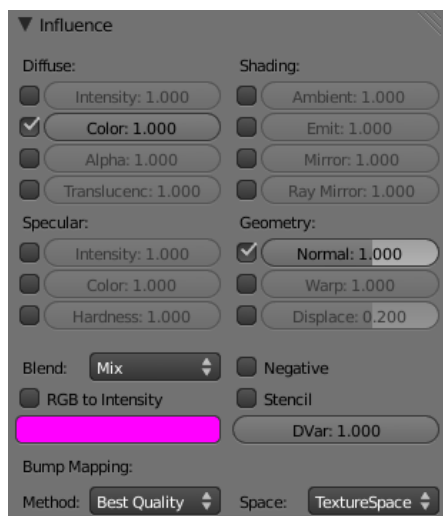
Spiele

UV Mapping ist auch essentiell in der Blender Game Engine, oder jedem anderem Spiel. Es ist der de facto Standard für das Anwenden von Texturen auf ein Model; so gut wie alle Modelle die Sie in einem Spiel finden, benutzen UV Mapping.

Material Texturen Beeinflussung

Not only can textures affect the color of a material, they can also affect many of the other properties of a material. The different aspects of a material that a texture influences are controlled in the Influence panel.

Oberflächen und (WIRE) Materialien



Texture Influence panel for a Surface material

Note

Texture options for Surface and Wire materials and in some cases also for Volume and Halo materials.

Diffuse

Intensity

Amount texture affects affects diffuse reflectivity

Color

Amount texture affect the basic color or RGB value of the material

Alpha

Influences the opacity of the material. See [Use Alpha for Object Transparency](#). Also use Z Transparency for light and if combining multiple channels.

Translucency

Influences the Translucency amount.

Specular

Intensity

Amount texture affect specular reflectivity

Color

Influences the Specular color, the color of the reflections created by the lamps on a glossy material.

Hardness

Influences the specular hardness amount. A DVar of 1 is equivalent to a Hardness of 130, a DVar of 0.5 is equivalent to a Hardness of 65.

Shading

Ambient

Influences the amount of Ambient light the material receives.

Emit

Influences the amount of light Emitted by the material.

Mirror

Influences the mirror color. This works with environment maps and raytraced reflection.

Ray Mirror

Influences the strength of raytraced mirror reflection.

Geometry

Normal

Commonly called bump mapping, this alters the direction of the surface normal. This is used to fake surface imperfections or unevenness via bump mapping, or to create reliefs.

Warp

Warp allows textures to influence/distort the texture coordinates of a next texture channel. The distortion remains active over all subsequent channels, until a new Warp has been set. Setting the factor at zero cancels out the effect.

Displace

Influences the Displacement of vertices, for using [Displacement Maps](#).

Blending

Blend

How this channel interacts with other channels below it. See [Compositing Mix Node](#) for information and examples on the effect of each mixing mode.

RGB to intensity

With this option, an RGB texture (affects color) is used as an intensity texture (affects a value).

Blend Color

If the texture is mapped to Col, what color is blended in according to the intensity of the texture? Click on the swatch or set the RGB sliders.

Negative

The effect of the Texture is negated. Normally white means on, black means off, Negative reverses that.

Stencil

The active texture is used as a mask for all following textures. This is useful for semitransparent textures and "Dirt Maps". Black sets the pixel to "untexturable". The Stencil mode works similar to a layer mask in a 2D program. The effect of a stencil texture can not be overridden, only extended. You need an intensity map as input.

DVar

Destination Value (not for RGB). The value with which the Intensity texture blends with the current value. Two examples:

- The Emit value is normally 0. With a texture mapped to Emit you will get maximal effect, because DVar is 1 by default. If you set DVar to 0 no texture will have any effect.
- If you want transparent material, and use a texture mapped to Alpha, nothing happens with the default settings, because the Alpha value in the Material panel is 1. So you have to set DVar to 0 to get transparent material (and of course Z Transparency also). This is a common problem for beginners. Or do it the other way round - set Alpha to 0 and leave Dvar on 1. Of course the texture is used inverted then.

Bump Mapping

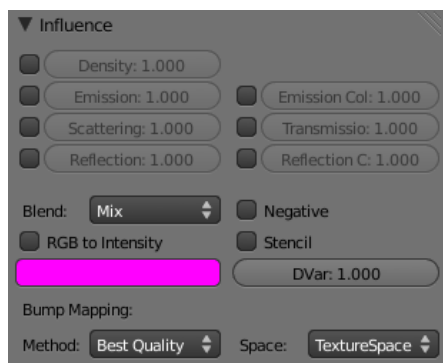
Settings for bump mapping.

Method

Best Quality, Default, Compatible, Original

Space

Texture Space, Object Space, View Space

Volume materials

Texture Influence panel for Volume material

Special texture options for Volume materials**Density**

Causes the texture to affect the volume's density.

Emission

Causes the texture to affect the volume's emission.

Scattering

Amount the texture affects scattering.

Reflection

Amount the texture affects brightness of out-scattered light

Emission Color

Amount the texture affects emission color.

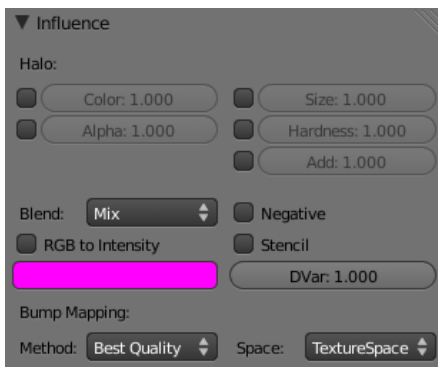
Transmission

Amount the texture affects result color after light has been scattered/absorbed.

Reflection Color

Amount the texture affects color of out-scattered light.

Halo materials



Texture Influence panel for a Halo material

Special texture options for Halo materials

Size

Amount the texture affects ray mirror.

Hardness

Amount the texture affects hardness.

Add

Amount the texture affects translucency.

Welt



World Panel

Blender stellt mehrere sehr interessante Einstellungen zur Verfügung, um Ihre Renderings zu vervollständigen, wie das Hinzufügen eines netten Hintergrunds und einigen interessanten 'Tiefen'-Effekten. Diese sind über den Kontexte zugänglich. Standardmäßig ist eine sehr einfache gleichförmige Welt gegeben. Sie können sie editieren oder eine neue Welt hinzufügen.

Zur Verfügung stehen:

[Background](#)

Die Farbe und Textur des Welthintergrunds, mit speziellen Einstellungen, um Koordinaten kartografisch darzustellen.

[Mist](#)

Fügen Sie einen Nebel zu Ihrer Szene hinzu, um das Gefühl von Tiefe zu erhöhen.

[Stars](#)

Bedeckt den Hintergrund zufällig mit Leuchtpunkten.

Während diese Welteinstellungen eine einfache Möglichkeit bietet, Effekte zu einer Szene hinzuzufügen, werden häufig [compositing nodes](#) bevorzugt. Diese sind zwar komplizierter, bieten aber zusätzliche Kontrolle und Optionen. Zum Beispiel kann der Z-Wert (Entfernung von der Kamera) gefiltert werden oder Normalen (Richtung von Oberflächen) können durch Compositing Nodes die Tiefe und räumliche Klarheit einer Szene vergrößern.

Hinweis

Einige der Einstellungen unter der Welt-Panel in Blender betreffen die Beleuchtung, dies finden Sie im [Lighting](#) Kapitel (siehe dazu [Ambient Light](#), [Exposure](#) und [Ambient Occlusion](#)). Wenn Sie eine Sun Lamp verwenden, sind Optionen für den Sky & Atmosphere im Lamp-Menü verfügbar.

Einführung

Rigging bezeichnet den Prozess ein Objekt so mit Funktionalität auszustatten das es nachher leicht animierbar ist. Es kommt aus dem (Puppen)Theater wo jede Puppe ein Gestell (Rigg) hat an dem es aufgehängt ist. Das bewegen der einzelnen Schnüre bewegt dann die Puppe. So ähnlich funktioniert das auch bei Animationen, nur das keine Schnüre und Gestelle gebraucht werden, stattdessen werden Knochen verwendet die man in die Objekte an die gewünschten Positionen setzt, mit dem Model verbindet und dann so bewegen kann wie man will. Deswegen gibt es neben dem Bearbeitungsmodus für Skelette auch einen Posiermodus in dem das Skelett und damit das Skelett in eine bestimmte Pose gebracht werden kann, ohne die ursprüngliche Position der Knochen und des Models zu verlieren. Diese Seite hat das Ziel alle diese und damit zusammenhängende Bereiche zu sammeln.

Skelette

Skelette sind die Objekte die zum Riggen benutzt werden. Ihr Name kommt von den vielen Gemeinsamkeiten mit echten Knochen.

[Skelette und Knochen Panele](#)

Hier ist erklärt was die unterschiedlichen Panele für Optionen bieten.

[Knochen](#)

Knochen sind die Grundbausteine von Skeletten.

[Visualisierung](#)

Es gibt vier verschiedene Darstellungsmodi die hier erklärt werden.

[Strukturen](#)

Strukturen in Skeletten bieten einige Vorteile. Ihre Benutzung und Funktion wird hier erklärt.

[Auswahl](#)

Das auswählen von den gewünschten Knochen kann manchmal schwierig sein.

Bearbeitung

[Knochen](#)

Lerne wie und mit welchen Auswirkungen man in Blender Knochen bearbeiten kann.

[Skizze](#)

Das Skizzier Werkzeug kann dazu benutzt werden Knochen zu skizzieren und einfach zu Skeletten zu machen.

[Vorlagen](#)

Vorlagen können nützlich sein wenn man viele gleichförmige Skelette benutzt und ein erstelltes Skelett wiederverwenden will.

Verbinden

Hier wird erklärt wie man Skelette mit Objekten verbindet.

[Verbinden von Objekten mit Knochen](#)

Es gibt mehrere Methoden Knochen und Objekte mit einander zu verbinden die hier erklärt sind.

[Verbinden von Objektformen](#)

[Retargeting](#)

Posen

[Visualisierung](#)

Es gibt verschiedene Visualisierungsmodi für Posen.

[Bearbeiten von Posen](#)

Hier ist erklärt wie Posen erstellt werden.

[Posen Bibliothek](#)

Posen können in einer Bibliothek gespeichert und dann wiederverwendet werden.

[Using Constraints](#)

Beschränkungen können den Arbeitsaufwand beim Animieren stark reduzieren.

[Inverse Kinematics](#)

Lerne hier über umgekehrte Kinematik Berechnung in Blender.

[Spline IK](#)

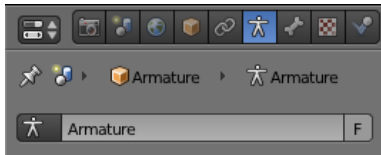
Ein Spezialfall der umgekehrten Kinematik.

Skelettpanel Übersicht

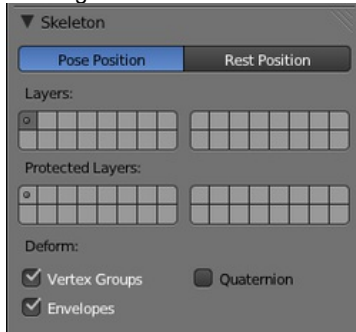
Mode: Objektmodus, Bearbeitungsmodus und Posenmodus

Panel: im Eigenschaften Fenster, Objekteigenschaften

Hier ist eine Übersicht über die verschiedenen Panele im Eigenschaftenfenster die die verschiedenen Eigenschaften von Skeletten steuern:



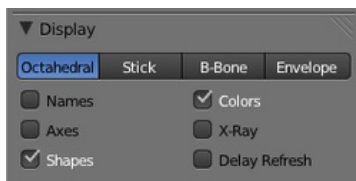
Das Eigenschaftenfenster.



Das Skelettpanel.

Skelettpanel in der Übersicht

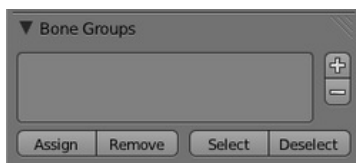
Dieses Panel ermöglicht die einfache Anordnung von Knochen in verschiedenen Ebenen.



Das "Display" Panel.

Darstellungspanel (alle Modi)

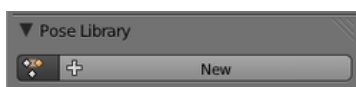
Dieses Panel kontrolliert die Darstellung der Knochen in der 3D Ansicht. Es gibt 4 unterschiedliche Darstellungsmodi.



Das Knochengruppenpanel

Knochengruppenpanel (Pose Modus)

Dieses Panel gibt eine Übersicht über Knochengruppen. Sie funktionieren ähnlich wie Objektgruppen.



Die Posen Bibliothek.

Posen Bibliothek (Posen Modus)

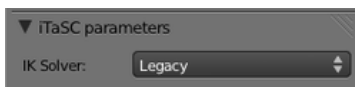
Erlaubt das Speichern von Posen um sie später wieder zu verwenden.



Das Geistpanel

Geistpanel (alle Modi)

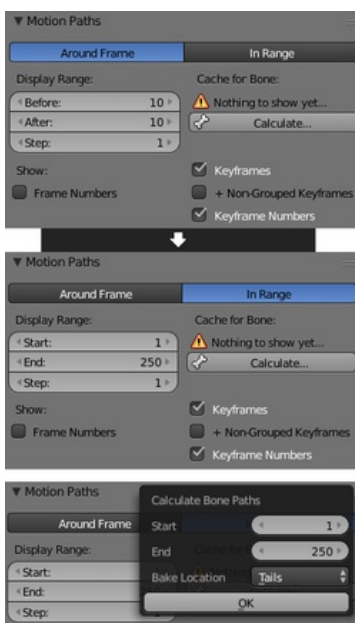
Erlaubt das Einstellen von aufeinander folgenden Posen.



Das iTaSC Parameterpanel.

"iTaSC" Parameterpanel (alle Modi)

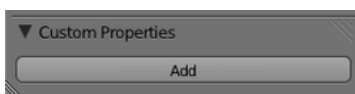
Defines the type of IK solver used in your animation.



Das Bewegungspfadpanel.

Bewegungspfadpanel (Posen Modus)

In diesem Panel können Visualisierungen für die Bewegungspfade eingeschaltet werden.



Das Benutzereigenschaftenpanel.

Benutzereigenschaftenpanel (alle Modi)

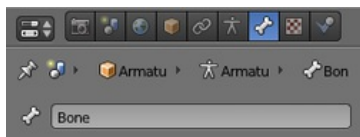
Panel für Benutzerdefinierte Eigenschaften die beim Skripten verwendet werden.

Knochenpanel Übersicht

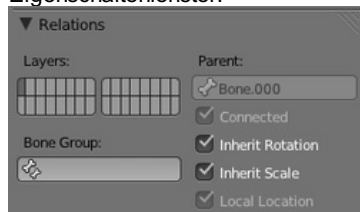
Mode: Objektmodus, Bearbeitungsmodus und Posenmodus

Panel: Alle im Eigenschaften Fenster, Knocheneigenschaften

Hier ist eine übersicht über die verschiedenen Optionen für Knocheneigenschaften:



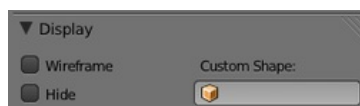
Die Knocheneigenschaften im Eigenschaftenfenster.



Das Beziehungspanel.

Beziehungspanel (Bearbeitungsmodus)

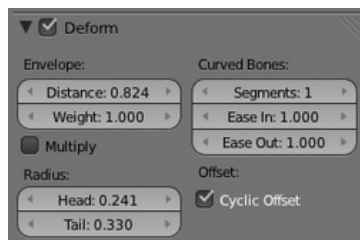
Hier können die Knoche auf verschiedenen Ebenen verschoben werden.



Das Darstellungspanel.

Darstellungspanel (Objektmodus)

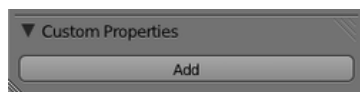
Hier kann die Darstellung von Knochen eingestellt werden.



Das Verformungspanel.

Verformungspanel (alle Modi)

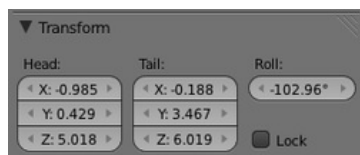
In this panel you can set basic properties of the bones.



The Custom Properties panel.

Benutzereigenschaftenpanel (alle Modi)

Panel für Benutzerdefinierte Eigenschaften die beim Skripten verwendet werden.



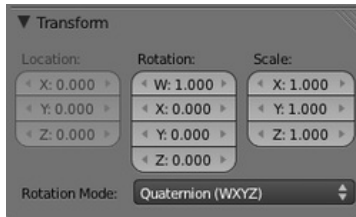
Das Transformationspanel

(Bearbeitungsmodus).

Transformationspanel (Bearbeitungs- und Posenmodus)

Im Bearbeitungsmodus kontrolliert dieses Panel die Position und Drehung von Knochen.

Im Posenmodus können hier Position, Rotation und Skalierung von Knochen festgelegt werden.



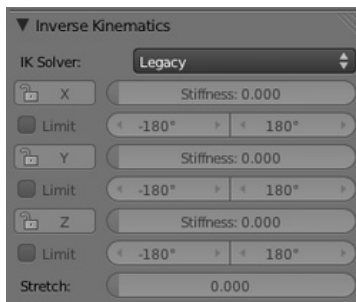
Das Transformationspanel (Posenmodus).



Das Transformationsbegrenzungspanel.

Transformationsbegrenzungspanel (Posenmodus)

Dieses Panel erscheint nur im Bearbeitungsmodus und begrenzt die maximalen Bewegungsmöglichkeiten von Knochen. Sehr nützlich zum Begrenzen von Knochenbewegung bei der Benutzung von UK.

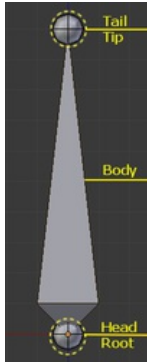


Das Panel für Umgekehrte Kinematik.

Panel für Umgekehrte Kinematik (Posenmodus)

Dieses Panel stellt ein wie sich Knochen bei umgekehrter Kinematik verhalten.

Knochen



Die Teile
eines
Knochens

Knochen sind die Grundelemente von Skeletten.

Sie haben drei Teile:

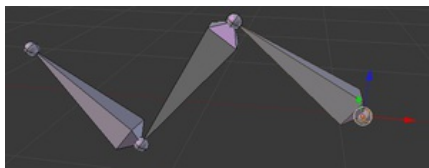
- den Startpunkt (engl. "root/head")
- den eigentlichen Knochen (engl. "body")
- den Endpunkt (engl. "tip/tail")

Im Bearbeitungsmods können Knochen ähnlich wie Vertices bewegt werden bis sie richtig positioniert sind. Allerdings können Knochen anderen Knochen ähnlich [wie Objekte](#) untergeordnet werden was das Bearbeiten entsprechend beeinflusst. Select the [\[\[Doc:2.5/Manual/Rigging/Armatures/Object mode#Your first\]\]](#)

Start und Endpunkt von Knochen definieren den Knochen durch ihre Position, abgesehen davon haben Knochen auch einen Radius der für die Umgebungsbeeinflussung benötigt wird.

Knochen Visualisierung

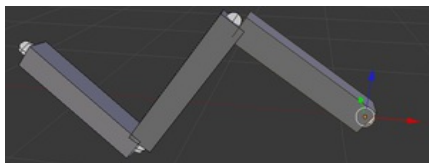
Knochen können auf verschiedene Arten in der 3D Ansicht dargestellt werden: als Oktahedron, in Stockform, als B-Bone, mit ihrem Einfluss und als Draht. Es können auch Benutzerdefinierte Formen verwendet werden.



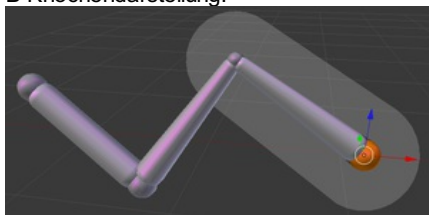
Oktahedrische Knochendarstellung.



Stock Knochendarstellung.



B Knochendarstellung.



Umgebungs Knochendarstellung.

Da Knochen Teile von Skeletten sind siehe [diese Seite](#) zu den Optionen.

Das Einschalten der Achsenbox im Darstellungspanel der Skeletteigenschaften zeigt die lokalen Achsen der Knochen an. Die Y-Achse verläuft dabei in Richtung des Knochens, von Start zum Endpunkt.



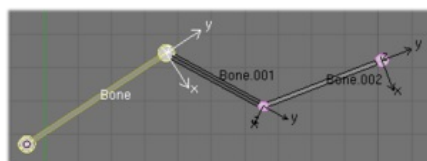
Der Knochenkontext

Knocheneigenschaften

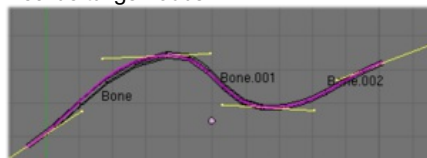
Im Bearbeitungs und Posenmodus sind die Knocheneigenschaften im Eigenschaftsfenster einsehbar. Diese bieten einige Informationen bzw. Einstellungen für den ausgewählten Knochen wie z.b. Name, übergeordnete Knochen und das Verformungsverhalten.

Knochen Steifigkeit

Obwohl Knoche absolut steif sind und sich nicht biegen, kann eingestellt werden das die einzelnen Knochen im Posen und Objektmodus als B Knochen oder Draht segmentiert dargestellt werden und wirken. Es werden dabei weiche Übergänge zwischen den lokalen y Richtungen der Knochen hergestellt.



Ein B Knochen Skelett im Bearbeitungsmodus.



Die Bezierkurve die über die Knochen gelegt wurde mit den Hebeln an den Knochenenden.



Das gleiche Skelett im Objektmodus

Beim Verknüpfen von Knochen zu einer Knochenkette berechnet Blender die Bezierkurve durch alle Knochenenden. Die Knochensegmente folgen dann dieser Kurve.

Diese Kurve kann allerdings nicht bearbeitet oder dargestellt werden. Einstellungen bezüglich des Verhaltens von Knochensegmenten müssen am Knochen vorgenommen werden.

Im Bild "Ein B Knochen Skelett im Bearbeitungsmodus" wurden 3 Knochen verknüpft die jeweils aus 5 Segmenten bestehen. Diese werden ebenfalls als B Knochen dargestellt.

Meshes die von segmentierten Knochen beeinflusst werden werden genauso weich deformiert wie die Segmente vermuten lassen.

Allerdings hat eine solche Knochenkette keinen Einfluss auf angehängte Objekte.

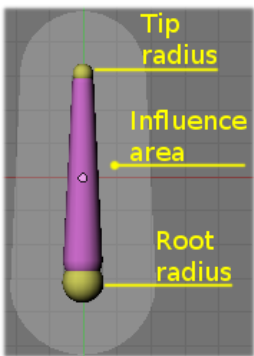
Wenn der Darstellungsmodus nicht auf B Knochen oder Draht eingestellt ist, werden keine Knochensegmente dargestellt obwohl sie immernoch da sind und wirken! ([Auf Objektdaten "skinnen"](#)).

Das bedeutet das auch in der Oktahedrischen Darstellungsweise Knochensegmente ein Mesh weich verformen können.

Knocheneinfluss

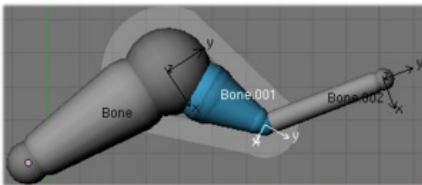
Ein Knochen bestimmt wie sich ihm zugeordnete Vertices verformen. Dafür ist es nötig zu bestimmen welche und wieviele Vertex zu einem Knochen gehören.

Der einfachste Weg das zu erreichen ist das Benutzen von "Knocheneinflüssen" und wird auch Umgebungsbeeinflusste Technik genannt (engl. "envelope technique"). Diese Methode ist standardmäßig aktiviert und verformt Vertices in einer einstellbaren Nähe vom Knochen mit einer entsprechenden Gewichtung.



Ein Knochen in Einflussdarstellung im Bearbeitungsmodus.

Wenn ein Knochen mit seiner Umgebung dargestellt wird ist der Einfluss des Knochens in Bearbeitungs und Posenmodus sichtbar. Dieser hängt ab von der eingestellten Entfernung und den Größen von Start und Endpunkt des Knochens.

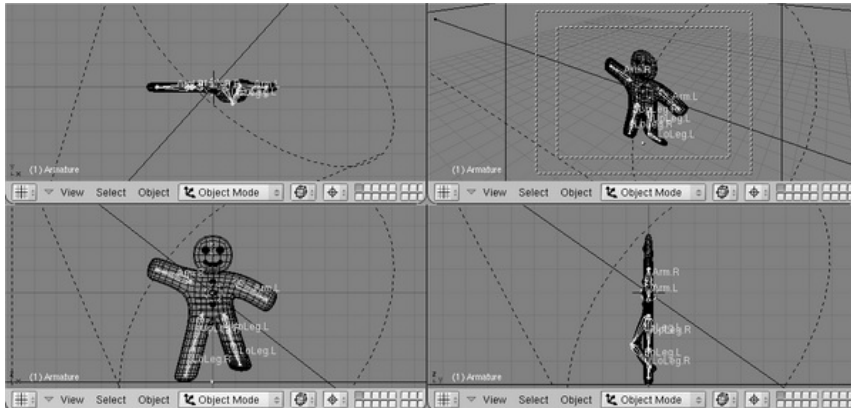


Ein Skelett in der Umgebungsdarstellung im Posenmodus.

All diese Einflussparameter sind im "Skinning" Teil diese Kapitels weiter behandelt.

Skinning

Nach dem auslegen eines passenden Skeletts ist es nötig dieses Skelett an ein Objekt bzw. Teile davon zu binden. Das wird in Blender "Skinning" genannt.



Das Lebkuchenmännchen verbunden mit einem Skelett.

Es gibt zwei Wege in Blender um ein Objekt mit einem Skelett zu verbinden:

- Es kann das Meshobjekt dem Skelettobjekt [untergeordnet](#) werden.
- Oder es kann der [Skelettmodifikator](#) benutzt werden.

Davon abgesehen sind einige Dinge zu beachten die auf den folgenden Seiten erklärt werden.

Posieren

Wenn Ihr Skelett von dem/den benötigten/benötigten Objekt/Objekten [skinned](#) ist, können Sie damit beginnen es zu Posen. Grundlegend, Deformieren oder Transformieren des Skin Objektes durch das Transformieren des Knochens. Aber dass tun Sie nicht im Editiermodus – denken Sie daran, dass Sie in diesem Modus die "Standard, Basis, "Rest" Position des Skelettes" bearbeiten. Sie können nicht den Objektmodus nutzen, da Sie damit nur ganze Objekte transformieren...

Also, steht für Skelette in Blender ein dritter Modus bereit, der Posen-Modus, der diesem Prozess zugeordnet ist. Es ist Art "Objektmodus für Knochen". In der rest Position (wie im Editiermodus bearbeitet), verfügt jeder Knochen über seine eigene Position/Rotation/Skalierung zu neutralen Werten (z.B. *0.0* für Position und Rotation, und *1.0* für die Skalierung). Wenn Sie daher einen Knochen im Posen-Modus bearbeiten, schaffen Sie einen Offset in seinen Transformier Eigenschaften, von seiner Rest Position aus - was ähnlich zu den [Form-Tasten von Meshes](#) ist.

Posen Abteilungsüberblick

In dieser Abteilung werden Sie folgendes kennen lernen:

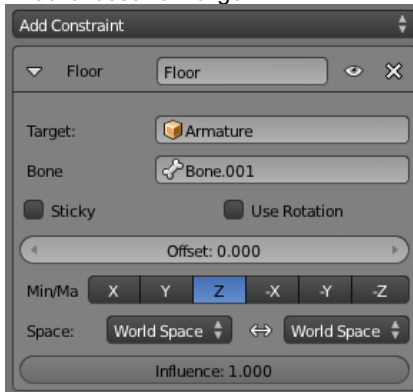
- Die [Visualisierungsfunktionen](#) spezifisch zum Posen-Modus.
- Das [Auswählen und Bearbeiten von Knochen](#) in diesem Modus.
- Das [Nutzen der Posen Bibliothek](#).
- Wie Sie [Beschränkungen nutzen](#) zum Kontrollieren der DoF (degrees of freedom = Freiheitsgrade) der Knochen.
- Wie sie [inverse Kinematik-Funktionen nutzen](#).

Auch wenn es für komplett statische Anlässe genutzt werden kann, ist das Posen stark mit den [Funktionen und Techniken zur Animation](#) verbunden. In diesem Teil, versuchen wir den Fokus auf Animation unabhängiges Posen zu legen, was aber nicht immer möglich ist. Wenn Sie also nichts über das Animieren in Blender wissen, wäre es eine gute Idee die ersten relevanten Kapitel (dem oberen Verweis folgend) durchzulesen, und anschließend hierher zurückzukehren...

Siehe

Wie gewöhnlich, können Sie sich dazu die [Tutorials](#) für Demonstrationen ansehen, und speziell [dieses BSoD](#).

Knochenbeschränkungen



Das Beschränkungspanel des Knochens Bone.001 im Posenmodus mit einer Boden Beschränkung

Da es nützlich ist Knochen manchmal als Objekte zu behandeln ist es möglich Knochen ähnlich wie Objekte zu [beschränken](#). Die Ähnlichkeiten gehen soweit das zu Beschreibung der Funktion auf die Beschränkungsseiten des Handbuchs verwiesen werden kann.



Dazu gehören bekannte Beschränkungen die zum Beispiel zum animieren von Maschinen benutzt werden können und Knocheneigene Beschränkungen wie die umgekehrte Kinematik.

Umgekehrte Kinematik

Umgekehrte Kinematik kehrt die "normale" Animationserzeugung um. Mit UK werden die Positionen von Knochen nicht durch ihre übergeordnete Knochen bestimmt, sondern der gewünschte Knochen wird positioniert und alle übergeordneten werden so positioniert, dass sie die eingestellte Position ermöglichen. Zum Beispiel kann so eine Hand platziert werden und alle Arm und Schulterknochen drehen und knicken so das die Hand position möglich ist.

Dadurch wird das einstellen von komplizierten Posen leichter.

Automatische UK

Automatische UK ist ein Werkzeug für schnelle Positionierung. Es kann in der Werkzeugleiste in der 3D Ansicht im Posenmodus eingeschaltet werden. Wenn die Auto UK eingeschaltet ist, werden standardmäßig alle übergeordneten Knochen so positioniert das sie die Position des ausgewählten und positionierten Knochens ermöglichen. Die Anzahl der dem ausgewählten Knochen übergeordneten Knochen (der übergeordneten Knochenkette) kann mit: ⌘ Shift PageUp, ⌘ Shift PageDown oder ⌘ Shift WheelUp , ⌘ Shift WheelDown  verändert werden.

Dies ist eine begrenzte Version der UK Begrenzung die aber trotzdem nützlich sein kann.

UK Beschränkungen

Meistens wird UK über Knochenbeschränkungen eingestellt. Siehe dafür die einzelnen Seiten zu:

- [UK Auflösung](#)
- [Spline UK](#)

Spline UK

Spline UK ist eine Beschränkung die alle Teile der eingestellten Knochenkette entlang kubischer Funktionen ausrichtet. Durch Veränderung der Biegesteifigkeit können ästhetische Formen erzeugt werden, besonders für organische Objekte wie Schwänze oder Tentakeln.

Einstellung

Die Spline UK Beschränkung ist nicht direkt eine umgekehrte Kinematik da sie strenggenommen eine Hilfestellung ist, um Knochen normal zu bewegen. Allerdings werden trotzdem mehrere Knochen bewegt und sie hat auch sonst einige Eigenschaften die sie als UK qualifizieren. Obwohl es vermieden werden sollte dazu zu greifen wird, wenn eine normale UK und eine Spline UK ausgewertet werden sollen die normale UK Priorität.

Siehe die Seite zur [Spline UK](#) für die Einstelloptionen.

Tips

- Für optimale deformation wird empfohlen das die Knochen ungefähr ähnliche Längen und nicht zu lang sein sollten, damit sie sich der Kurve gut anpassen können. Außerdem sollten die Knochen in der Ruheposition ungefähr der Kurve folgen.
- Für bessere Kontrolle über die Kurve wird empfohlen das Haken zur veränderung der Kurve benutzt werden sollten.
- Es ist im Moment nötig das Kontrollknochen in einem separaten Skelett sind, da Blender sonst in Abhängigkeitskonflikte gerät.

Beschränkungen

Beschreibung

Beschränkungen sind Objektfunktionen, sie kontrollieren auf verschiedene Weise *Transformationseigenschaften* (Position, Rotation und Skalierung) eines Objektes.

Alle Beschränkungen teilen sich eine einfache [gemeinsame Oberfläche](#), wieder mit vielen Ähnlichkeiten zu denen von Modifikatoren.

Benutzung von Beschränkungen

Auch wenn Beschränkungen in statischen Szenen sehr nützlich sein können, sind sie eigentlich für Animationen ausgelegt worden, das sie die Kontrolle von Freiheitsgraden von Objekten sehr leicht einstellbar begrenzen können.

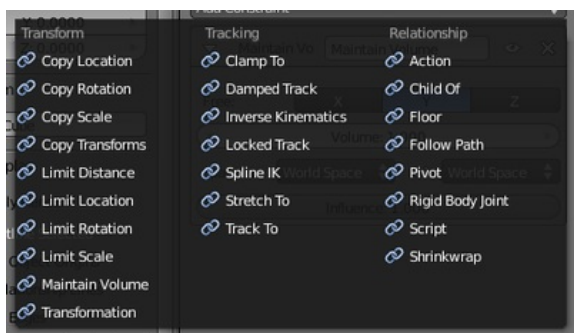
Beschränkung von Knochen

Im Posiermodus von Knochen verhält sich jeder Knochen ein ungefähr wie ein normales Objekt, daher können sie auch beschränkt werden. Die meisten Beschränkungen funktionieren gut mit Objekten und Knochen aber es gibt ein paar Ausnahmen die auf der jeweiligen Seite erklärt werden.

Verwandte Themen sind:

- [Das Kapitel über Animationen](#)
- Das benutzen von Beschränkungen in Verbindung mit [Armatures](#)

Verfügbare Beschränkungen



Beschränkungs Menü

Es gibt verschiedene Typen von Beschränkungen die in 3 Klassen unterteilt werden:

Page status ([reviewing guidelines](#))

Partial page Text

Missing the Motion Tracking category of constraints

Images

Old Screenshot

Add Constraints list missing Motion Tracking .
25-Manual-Constraints-Menu.png

Proposed fixes:

The category for Motion Tracking constraints needs to be written
Image needs updating

- [Transformationsbeschränkungen](#)
- [Beschränkungen durch Folgen](#)
- [Beziehungsbeschränkungen](#)

Es gibt Beschränkungen die nur mit dem Objekt das sie hat funktionieren und es gibt welche die ein zweites Zielobjekt benötigen, manchmal von einem bestimmten Objekttyp. Beziehungen zwischen Eigentümer der Beschränkung und Zielobjekt werden mit einer dunkelblauen gestrichelten Linie dargestellt.

Transformationsbeschränkungen

Diese Beschränkungen begrenzen die Freiheitsgrade des Eigentümers, entweder absolut (also in globalen Koordinaten) oder relativ zu einem Zielobjekt

[Kopiere Transformationen](#)
[Kopiere Position](#)

Kopiert die Transformationen des Zielobjektes.

Kopiert die Position des Zielobjektes mit einem optionalen Abstand, sodass beide zusammen bewegt werden.

Kopiere Ausrichtung	Kopiert die Ausrichtung des Zielobjektes mit einem optionalen Abstand, sodass beide zusammen gedreht werden.
Kopiere Skalierung	Kopiert die Skalierung des Zielobjektes mit einem optionalen Abstand, sodass beide zusammen skaliert werden.
Beschränke Entfernung	Beschränkt den Abstand vom Eigentümer zum Zielobjekt auf "näher als" oder "genau" oder "weiter als" eine angegebene Entfernung.
Beschränke Position	Beschränkt die Position des Eigentümers.
Beschränke Ausrichtung	Beschränkt die Ausrichtung des Eigentümers.
Beschränke Skalierung	Beschränkt die Skalierung des Eigentümers.
Transformation	Benutzt den Wert einer Eigenschaft des Ziels (Position, Ausrichtung, Skalierung) zur Bestimmung eines Eigenschaftswerts des Eigentümers.
Volumen beibehalten	Hält das Volumen eines Objekts konstant.

Beschränkung durch Folgen

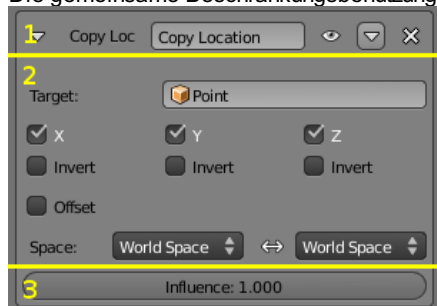
Diese Beschränkungen versuchen den Eigentümer so zu beeinflussen das er dem Ziel folgt.

Klammern	Der Eigentümer klammert sich an ein Kurvenobjekt.
Damped Track	
Umgekehrte KinematikInverse Kinematics	Ist nur bei Knochen möglich. Erzeugt eine Knochenkette die vom Ziel kontrolliert wird.
Festgelegte Bahn	Der Eigentümer folgt einem Ziel, aber mit einem beschränkten Freiheitsgrad.
Spline Umgekehrte Kinematik	Benutzt mit kubischen Funktionen angenäherte umgekehrte Kinematik.
Strecken	Streckt das Objekt zum Ziel.
Folgen	Die Ausrichtung des Objekts folgt dem Ziel.

Beziehungsbeschränkungen / Andere

Aktion	Das Objekt führt eine Aktion vom Ziel abhängig aus.
Untergeordnet	Erlaubt eine Beschränkung der Auswirkungen von Unterordnungen.
Boden	Benutzt die Position des Ziels zum Festlegen eines Bodens oder einer Wand die das Objekt nicht passieren kann.
Folge Pfad	Das Objekt folgt einem Pfad oder einer Kurve.
Pivot	
Feste Körperverbindung	Erzeugt eine feste Verbindung zwischen Objekt und Ziel.
Skript	Benutzt ein Pythonskript das die Beschränkung festlegt.
Schrumpfwickeln	Beschränkt die Position des Objekts unter anderem auf die Oberfläche des Ziels.

Die gemeinsame Beschränkungsbenutzungsoberfläche



Die drei Teile von Beschränkungen

Genau wie bei [Modifikatoren](#), kann ein Objekt oder [Knochen](#) mehrere Beschränkungen auf einmal benutzen. Deswegen werden sie in einer Hierarchie geordnet die ihren Einfluss festlegt.

Alle Beschränkungen haben eine gemeinsame Oberfläche:

1. Der Kopf der die meisten allgemeinen Einstellungen enthält
2. Die beschränkungsspezifischen Einstellungen.
3. Der Einfluss und die Animationskontrollen

Der Kopf



ein Beschränkungskopf

Hier sind zu finden:

Ein kleiner Pfeil der die Einstellungen der Beschränkung verstecken und wieder hervorholen kann.

Der Beschränkungstyp

Das Namesfeld wo ein Name für die Beschränkung eingegeben werden kann.

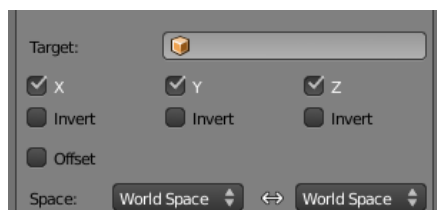
Außerdem wird es rot unterlegt wenn die Beschränkung nicht funktioniert, sei es weil kein Ziel angegeben ist oder weil die Einstellungen fehlen oder widersprüchlich sind.

Ein Auge das festlegt ob die Beschränkung angezeigt wird.

Die hoch und runter Knöpfe die die Position der Beschränkung in der Hierarchie verändern.

Das X zum löschen der Beschränkung.

Beschränkungseinstellungen



Der Hauptteil einer Beschränkung enthält Einstellungen, das Ziel und den Beschränkungsraum.

Die Beschränkungseinstellungen haben natürlich spezifische Auswirkungen abhängig vom Typ. Allerdings gibt es auch hier Gemeinsamkeiten.

Das Ziel

Hier wird das Ziel angegeben oder ausgewählt(siehe [Beschränkungen](#)).

Bei einem Mesh- oder Netzobjekt erscheint zusätzlich ein Feld für eine Vertexgruppe auf die der Einfluss der Beschränkung eingegrenzt werden kann.

Ähnlich wird bei einer Armatur ein Knochenfeld verfügbar. Bei Angabe eines Knochens ist ein Skalierungsfaktor für den Punkt des Knochens der benutzt werden soll von 0 (Wurzel des Knochens) bis 1 (Ende des Knochens) verfügbar.

Der Beschränkungsraum

Für viele Beschränkungen gibt es die Möglichkeit einen Raum anzugeben in dem die Beschränkung angewendet wird. Man kann hier auswählen welches lokale bzw. ob das globale Koordinatensystem zur Bestimmung von Punkten und Einflüssen benutzt wird. Es sind verfügbar:

Lokale Koordinaten

Operationen werden im Bezug auf den Ursprung des des Objekts berechnet.

Lokale mit Übergeordneten (nur für Knochen)

Auswirkungen werden auf den lokalen Koordinaten des übergeordneten Knochens errechnet.

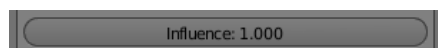
Posierungsraum (nur für Knochen)

ermöglicht den Einfluss der Pose ohne einen Einfluss des normalen Raums der 3D Ansicht im Objekt oder Bearbeitungsmodus.

Globale Koordinaten (standard)

Operationen werden im Bezug auf den Ursprung des 3D Raums berechnet.

Einfluss



Einfluss

Der Einfluss skaliert von 0 bis 1 wie stark sich die Beschränkung auswirkt.

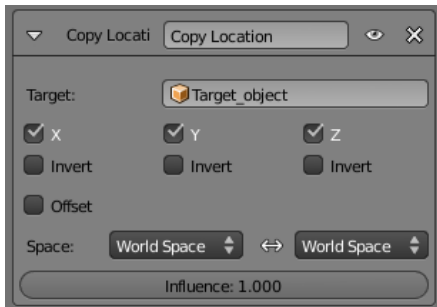
Kopiere Position

Beschreibung

Die Kopiere Position Beschränkung setzt die Position des Eigentümerobjekts auf die des Ziels oder relativ dazu.

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



Kopiere Position Panel

Abstand

Stellt den Abstand zur kopierten Position ein.

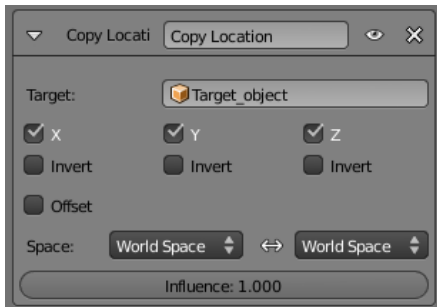
Kopiere Rotation

Beschreibung

Die Kopiere Rotation Beschränkung setzt die Rotation des Eigentümerobjekts auf die des Ziels oder relativ dazu.

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



Kopiere Position Panel

Unterschied

Stellt den Unterschied zur kopierten Rotation ein.

Kopiere Skalierung

Beschreibung

Die Kopiere Skalierung Beschränkung setzt die Skalierung des Eigentümerobjekts auf die des Ziels oder relativ dazu.

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



Kopiere Position Panel

Unterschied

Stellt den Unterschied zur kopierten Skalierung ein.

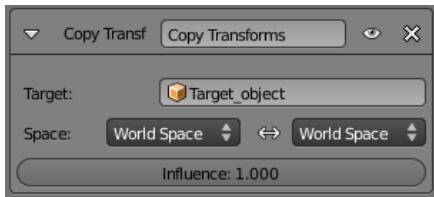
Transformationen Kopieren

Beschreibung

Diese Beschränkung kopiert alle Transformationen des Ziels für das Objekt.

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



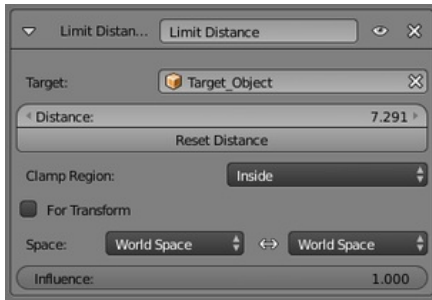
Copy Transforms panel

Beschränke Entfernung

Beschreibung

Diese Beschränkung zwingt das Eigentümer Objekt in einer bestimmten Entfernung zum Ziel zu sein.

Optionen



Limit Distance panel

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).

Entfernung

Hier kann die Entfernung eingegeben werden.

Entfernung zurücksetzen

Setzt die Entfernung auf die Entfernung zwischen Objekt und Ziel.

Region

Stellt ein wie die Entfernung angewendet werden soll:

innerhalb (standard)

Das Objekt bleibt innerhalb der Entfernung zum Ziel.

außerhalb

Das Objekt bleibt außerhalb der Entfernung zum Ziel.

exakt

Das Objekt bleibt genau auf der Entfernung zum Ziel.

Position begrenzen

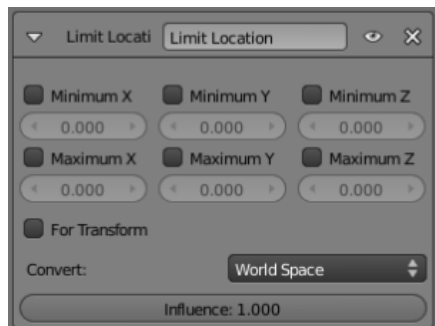
Beschreibung

Objekte können entlang der X Y oder Z Achse bewegt werden. Diese Beschränkung begrenzt diese Bewegungen durch untere und obere Grenzen.

Blenderintern wird die Position aber trotzdem immernoch verändert, es wird nur nicht mehr dargestellt oder gerendert.

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



Limit Location panel

Minimum X, Minimum Y, Minimum Z

Diese Knöpfe aktivieren die Einstellung für die unteren Grenzen.
Im Feld ist der Wert einzugeben.

Maximum X, Maximum Y, Maximum Z

Diese Knöpfe aktivieren die Einstellung für die oberen Grenzen.
Im Feld ist der Wert einzugeben.

Für Transformation

Setzt auch die Blenderinternen Koordinaten auf einen durch die Beschränkung zugelassenen Wert.

Rotation begrenzen

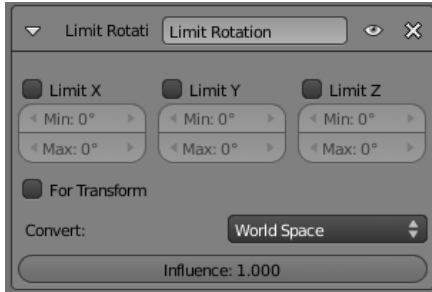
Beschreibung

Objekte können um die X Y oder Z Achse gedreht werden. Diese Beschränkung begrenzt diese Rotation durch untere und obere Grenzen.

Blenderintern wird die Rotation aber trotzdem immernoch verändert, es wird nur nicht mehr dargestellt oder gerendert.

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



Limit Rotation panel

Begrenze X, Begrenze Y, Begrenze Z
begrenzt die Rotation des Objekts auf
Die Minnimum und Maximum Werte.

- Anders als bei der [Positionsbegrenzung](#), können nicht obere und untere Grenzen unabhängig eingestellt werden.

Für Transformation

Setzt auch die Blenderinternen Koordinaten auf einen durch die Beschränkung zugelassenen Wert.

Skalierung begrenzen

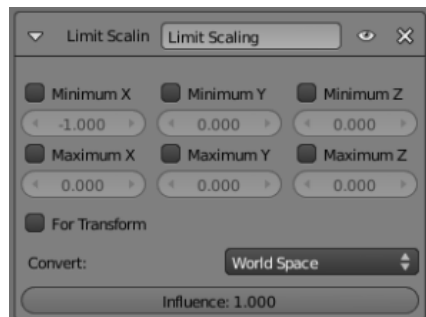
Description

Objekte können entlang der X Y oder Z Achse skaliert werden. Diese Beschränkung begrenzt diese Bewegungen durch untere und obere Grenzen.

Blenderintern wird die Position aber trotzdem immernoch verändert, es wird nur nicht mehr dargestellt oder gerendert.

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



begrenze Skalierung Panel

Minimum X, Minimum Y, Minimum Z

Diese Knöpfe aktivieren die Einstellung für die unteren Grenzen.
Im Feld ist der Wert einzugeben.

Maximum X, Maximum Y, Maximum Z

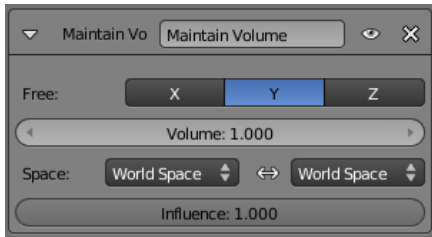
Diese Knöpfe aktivieren die Einstellung für die oberen Grenzen.
Im Feld ist der Wert einzugeben.

Volumen beibehalten

Beschreibung

Beschränkt das Objekt auf eine Skalierung seines Ursprünglichen Volumens.

Optionen



Volumen beibehalten Panel

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).

Frei X/Y/Z

Das freie Skalieren eines Objekts.

Volumen

standard ist 1.0.

Siehe auch

- (Englisch)[Harkyman on the development of the Maintain Volume constraint](#), März 2010

Transformations Beschränkung

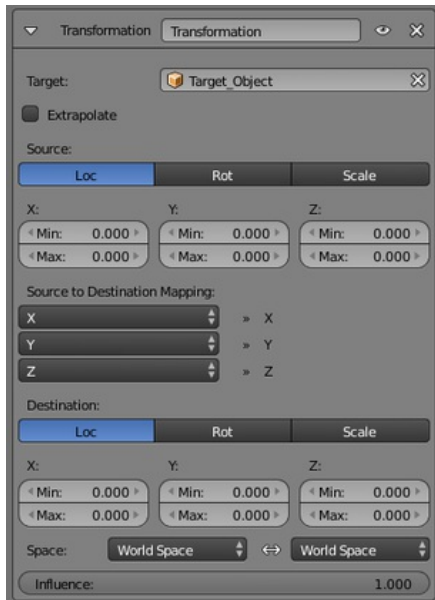
Beschreibung

Diese Beschränkung ist die Kombination von den Kopier- und Begrenzungsbeschränkungen. Sie kopiert die Transformation des Ziels auf den Eigentümer, aber mit vielen Konfigurationsmöglichkeiten wie welche Werte des einen in die Werte des anderen kopiert werden sollen.

Es kann die Verschiebung des Ziels auf der X Achse für das Objekt eine Verschiebung auf der Z Achse verursachen. Aber es ist auch möglich die Rotation um die Y auf Bewegung entlang der X Achse zu "übersetzen".

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



Transformations Panel

Extrapolieren

Normalerweise werden nur Bewegungen innerhalb der angegebenen Grenzen auf das andere Objekt übertragen. Durch aktivieren des Extrapolierens verursachen Veränderungen außerhalb dieser Grenzen trotzdem Veränderungen am Objekt, es verhält sich dann so als ob die Veränderung relativ wie in den Grenzen wäre. Siehe das Bild unten als Beispiel: die Gerade steigt weiter obwohl die Grenzen sie eigentlich davon abhalten würden.

Quelle

Ist die Quelle der Transformationsinformation. Z.B. ein Hebel oder ein Rad.

Die drei Position, Rot und Skalierung Knöpfe stellen ein welcher Typ von Eingabewerten benutzt wird.

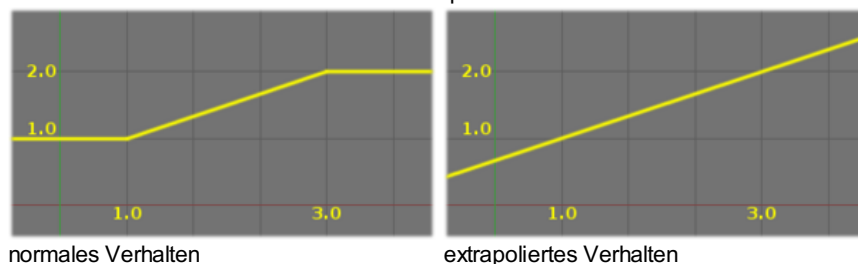
Die X:, Y: und Z: min und max Werte stellen die Grenzen ein innerhalb denen die Werte übertragen werden oder die als Grenzen zum Extrapolieren benutzt werden.

Ziel

Ist das Ziel also das Objekt das verändert werden soll.

- Die drei Position, Rot und Skalierung Knöpfe stellen ein welcher Typ von Eigenschaft im Zielobjekt verändert werden soll.
- Die drei Achsenübertragung Drop-downlisten erlauben welche Achse auf welche Achse übertragen wird (z.b. X nach Z).
- Die min und max Werte geben an welche Grenzen für die Veränderung des Zielobjekts existieren sollen. Das ist nicht das gleiche wie eine Begrenzung der Eingabewerte!

Das Extrapolieren



UK Auflösung

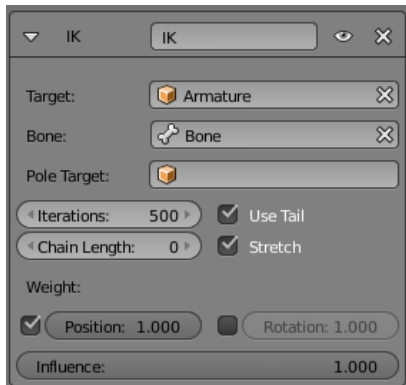
Beschreibung

Benutzt die "umgekehrte Kinematik" Animationstechnik, ansatt z.b für einen Finger den ganzen Arm einstellen zu müssen, richtet sich der Arm nach dem Finger aus. Nur für Knochen verfügbar.

Die Funktionsweise dieser Beschränkung ist außerdem auf der Seite für [umgekehrte Kinematik](#) vollständig dokumentiert.

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



Inverse Kinematics panel

Ziel

Muss eine Armatur sein.

Knochen

Ein Knochen in der Armatur.

Pol Ziel

Objekt für Polare Rotation.

Lösungsschritte

Maximale Anzahl an Lösungsschritten.

Kettenlänge

Wieviel Knochen von der Beschränkung beeinflusst werden (0 für alle).

Benutze Schwanz

Benutzt den Schwanz des Knochens als letztes Kettenelement.

Streckung

Erlauben UK Streckung.

Einfluss

Der Einfluss der UK auf die Knochenpositionen.

Position

Für eine UK in Baumstruktur: Einfluss auf eine Position in der Kette.

Rotation

Kette folgt der Rotation

Ziel

Objekt für das UK aufgelöst wird (normalerweise ein leeres Objekt) leer lassen um UK auszuschalten.

kubische UK

Beschreibung

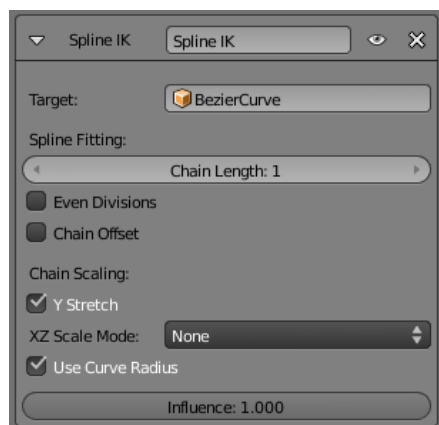
Der kubische UK lässt eine Knochenkette einer kubischen Funktion (engl. Spline) folgen. Es ist besonders geeignet für biologische und Runde Knochenkettten wie z.B. Schwänze und Seile.

Um UK anzuwenden ist ein bestimmter Aufbau notwendig: Man braucht eine Knochenkette und eine Kurve der die Knochen folgen sollen.

1. Dem letzten Knochen in der Kette wird diese Beschränkung auferlegt.
2. Die Kettlänge entspricht der Anzahl an Knochen in der Kette vom letzten Glied an die beeinflusst werden sollen.
3. Das Ziel ist die Kurve der die Knochen folgen sollen.

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



kubische UK

Ziel

Die gewünschte KurveThe target curve

Spline Fitting

Kettenlänge

Wieviele Knochen vom letzten an beeinflusst werden sollen.

gleichmäßige Unterteilung

Ignoriert die relative Länge der Knochen.

Ketten Abstand

legt den Abstand der Kette von der Kurve fest.

Kettenskalierung

Y Streckung

Streckt die Knochen um die Y Achsen beim anpassen an die Kurve

XZ Skalierungsmodus

Skaliert stattdessen entlang den Achsen X und Y.

Keine

keine Skalierung

Knochen Original

benutze die Ursprüngliche Skalierung der Knochen.

Volumenerhaltung

die Skalierung entlang der Achsen X und Z ist die Umkehrung der Skalierung entlang Y.

Benutze Kurvenradius

Durchschnittlicher Radius der Endpunkt wird benutzt um X und Z Skalierung einzustellen, zusätzlich zu X Z Skalierungsmodell.

Siehe auch

[Skelettaufbau und Positionierung](#).

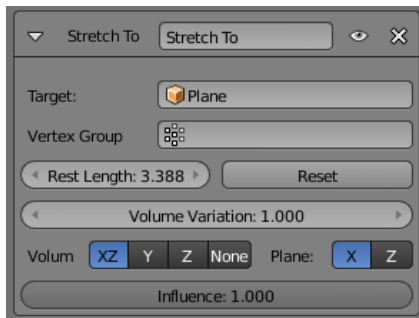
- Das release Log für die kubische UK (englisch) [Blender.org 2.56 Release Log for Spline IK](#)

Strecken

Diese Beschränkung funktioniert wie das [Verfolgen](#) aber mit einer zusätzlichen Eigenschaft. Das Objekt wird entlang der Y Achse die dem Ziel folgt skaliert, also zum Ziel gestreckt oder vom Ziel zusammengedrückt.

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



Strecken Panel

Ruheentfernung

Dieses Feld bestimmt die Ruheentfernung bei der keine Streckung oder Stauchung vorgenommen wird.

Zurücksetzen

Setzt die Ruheentfernung auf die aktuelle Entfernung vom Objekt zum Ziel und setzt dementsprechend die Verformung zurück.

Volumen VeränderungVariation

Diese Feld kontrolliert die Volumenveränderung proportional zur Streckungslänge.

Volumen

Diese Buttons bestimmen entlang welcher Achse skaliert werden soll um das Volumen annähernd zu erhalten.

Ausrichtung

Diese Einstellung ist Äquivalent mit der "Oben" Einstellung im der Verfolgen Beschränkung. Es sorgt dafür das die ausgewählte Achse nach Möglichkeit in Richtung der entsprechenden globalen Achse zeigt.

Verfolgen

Beschreibung

Diese Beschränkung sorgt dafür das die ausgewählte Achse des Objekts immer auf das Ziel gerichtet ist. Wird das Ziel bewegt, dreht sich das Objekt entsprechend. Es gibt eine Achse die dabei nach Möglichkeit nach oben zeigen sollen, so gut es eben geht.

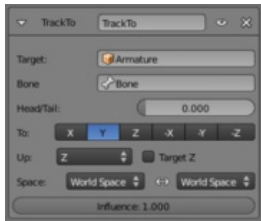
Sie ist dem [UK Auflöser](#) in Funktion sehr ähnlich.



"Billboard tracking" Tafel verfolgen

Ist sehr beliebt in der Spiele Industrie da so Tafel immer auf die Kamera ausgerichtet werden können. Wenn die Tafeln entsprechende Texturen haben können 3D Effekte erzeugt werden ohne die dafür sonst nötige Rechenleistung zu benötigen. Es wird häufig für Rauch und Nebel benutzt die sonst sehr aufwendig wären.

Optionen



Verfolgen Panel

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).

Nach

Die verfolgende Achse die das Objekt nach dem Ziel ausrichtet.

Oben

Die Achse die nach Möglichkeit nach oben zeigen soll.

Z des Ziels

Aktiviert das die nach oben zeigende Achse nicht global sondern relativ zum Ziel nach "oben" zeigt.

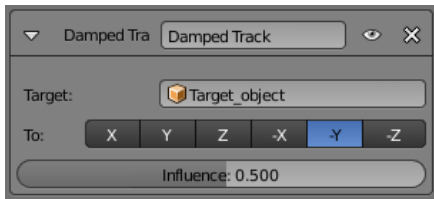
abgeschwächtes Verfolgen

Beschreibung

Diese Beschränkung lässt eine Achse immer auf das Ziel zeigen.

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



abgeschwächtes Verfolgen

Achse

Die Achse bzw. Ausrichtung die auf das Ziel zeigen soll

begrenztes Verfolgen

Beschreibung

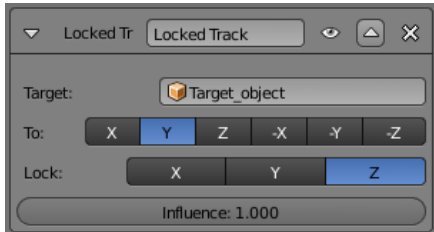
Das begrenzte Verfolgen funktioniert genau wie das einfache [Verfolgen](#) mit dem Unterschied das Objekt sein Ziel nicht irgendwie verfolgt, sondern dabei bestimmte Achsen festgelegt werden um die dabei nicht gedreht werden darf z.B. ein Kompass kann nur um die Achse drehen auf der er aufgehängt ist, nicht um die anderen 2.

Dabei kann es Helfen die Achsen für die Objektausrichtung einzuschalten oder den 3D Manipulator auf lokal einzustellen.

Weiter Dokumentation gibt es im (englischen) [2.49 BSoD verfolgen tutorial](#).

Optionen

Für allgemeine Optionen siehe [Beschränkungseinstellungen](#).



begrenztes Verfolgen Panel

Nach

Die Achse die das Ziel Verfolgen soll.T

Sperre

Sperrt rotation um die ausgewählte Achse.

Animation

Einleitung

Die Animation versetzt ein Objekt in Bewegung oder verändert seine äußere Form in zeitlicher Abhängigkeit. Objekte können auf viele Arten animiert werden:

Das Objekt selbst

kann in Position, Ausrichtung oder Größe verändert werden;

Dessen Verformung

erfolgt durch Animierung der Eck- oder Kontrollpunkte;

Figuren-Animation mit Hilfe von Skelettstrukturen

ermöglicht das Verformen auf Grund des Bewegens von "Knochen" innerhalb des Meshs, einer sehr komplexen und flexiblen Interaktion, die figürliche Objekte gehen und springen lässt.

In diesem Kapitel werden die ersten zwei Arten behandelt, die als Grundlage für das Verstehen der nachfolgenden Kapitel dienen.

Normalerweise werden in Animationsanwendungen drei Methoden verwendet, um 3D-Objekte zu bewegen:

Schlüsselbilder (Key-Frames)

Die kompletten Positionsinformationen zu einem bestimmten Zeitpunkt werden in Schlüsselbildern (Key-Frames) gespeichert. Eine Animation wird durch die flüssige Interpolation der Objektbewegung zwischen den einzelnen Schlüsselbildern erzielt. Der Vorteil dieser Methode ist es, mit klar visualisierten Elementen zu arbeiten. Ein Animator kann sich von einer Position zur nächsten vorarbeiten. Es ist ihm möglich, zuvor erstellte Positionen räumlich oder zeitlich zu verschieben.

Animatorkurven

Kurven werden von Schlüsselbildern interpoliert. Sie können einzeln für den X-, Y- und Z-Wert der Lage, Drehung und Größe, sowie für jedes andere Attribut in Blender erstellt werden. Diese bilden die Graphen für die Bewegung, mit der horizontalen Zeit- und der vertikalen Werteachse. Der Vorteil dieser Methode ist, dass die Ergebnisse der Bewegung genau kontrolliert werden können.

Pfad

Eine im 3D-Raum gezeichnete Kurve, die das Objekt in einer vorgegebenen Zeit folgen muss.

The first two systems in Blender are completely integrated in a single one, the [F-Curve system](#).

In Blender 2.5x, everything can now be animated. Previously, only certain datablocks had the ability to be keyframed. Now users have the ability to animate nearly any type of data that can be changed to multiple values.

Chapters

General Principles and Tools

[Keyframes](#)
[Animation Editors](#)
[Using The Timeline](#)
[Markers](#)

The Graph Editor

[Using the Graph Editor](#)
[F-Curves](#)
[Editing Curves](#)
[F-Curve Modifiers](#)

The Action Editor

[Actions](#)
[Creating Actions](#)

Animation Techniques

[Constraints](#)
[Moving objects on a Path](#)
[Changing Object Layers](#)
[Game Engine Physics Recording](#)

Animating Deformation

[Methods of deformation](#)
[Shape Keys](#)
[Deforming by a Lattice](#)
[Deforming with Hooks](#)

See also [Hooks](#) - Uses a modifier as a way to change the shape of a mesh. Sorta like sticking a fish hook in a mesh and pulling. Uses the principles discussed in Shape Keys.

Drivers

[Drivers](#)
[Driven Shape Keys](#)

The [BSoD Introduction to Character Animation tutorial](#) is a good starting point for learning character animation. Even if you never used Blender before.

Animation Basics

[Actions](#)

Actions are used to record the animation of objects and properties.

[Drivers](#)

Drivers are used to control and animate properties.

[Keying Sets](#)

Keying Sets are used to record a set of properties at the same time.

[Markers](#)

Markers are used to mark key points/events within an animation.

[Motion Paths](#)

Motion Paths are used to visualize an animation.

[Shape Keys](#)

Shape Keys are used to deform objects into new shapes.

Animation Editors

[Timeline](#)

The Timeline Editor is a quick editor to set and control the time frame.

This also has some tools for animation.

[Graph Editor](#)

The Graph Editor is mostly used to edit the F-Curves and Keyframes for Channels and Drivers.

[Dope Sheet](#)

The Dopes Sheet contains a collection of animation editors.

[NLA Editor](#)

The NLA Editor is used to edit and blend Actions together.

Categories

[Modifiers](#)

Modifiers are automatic operations that affect an object in a non-destructive way.

With modifiers, you can perform many effects automatically that would otherwise be tedious to do manually.

[Rigging](#)

Rigging.

[Constraints](#)

Constraints are a way of connecting transform properties (position, rotation and scale) between objects.

[Physical Simulation](#)

This category covers various advanced Blender effects, often used to simulate real physical phenomena.

There is the Particle System for things like hair, grass, smoke, flocks.

Soft Bodies are useful for everything that tends to bend, deform, in reaction to forces like gravity or wind.

Cloth simulation, to simulate clothes or materials.

Rigid Bodies can simulate dynamic objects that are fairly rigid.

Fluids, which include liquids and gasses, can be simulated, including Smoke.

Force Fields can modify the behavior of simulations.

[Motion Tracking](#)

Motion tracking is a new technique available in Blender. It is still under development, and currently supports basic operations for 2D motion tracking, 3D motion tracking, and camera solution.

[Animation Scripts](#)

Addon scripts for animation.

[Rigging Scripts](#)

Addon scripts for rigging.

See Also

- [Manual#Animation](#)

Der Zeitstrahl

Das *Zeitstrahl* Fenster, erkennbar am Symbol einer Uhr, wird standardmäßig am Bildschirmende angezeigt.


Der Zeitstrahl ist nicht wirklich ein "Editor", mehr wie das *Outliner* Fenster, ein "informierendes" Fenster, mit ein Paar eingeschränkten Bearbeitungsfunktionen. features.

Hier erhalten Sie einen Überblick über den Animierungsteil Ihrer Szene: wie ist die momentane Zeit, entweder in Bildern oder Sekunden, wo befinden sich die Schlüsselbilder des aktiven Objektes, was sind die Start-, und End-Bilderpositionen Ihrer Animation, Marker, usw.....

Es hat VTR-ähnliche Kontrollknöpfe, zum Wiedergeben der Animation, Ändern der momentanen Bildposition und Bildreichweite, scrollen zwischen den Schlüsselbildern ("VTR" steht für **V**ideo **T**ape **R**ecorder, soviel wie Video Band Aufnahme) .

Zeitstrahl Elemente

Das momentane Bild

The current frame is materialized by the thick green vertical line (the so-called "time cursor", in *Timeline window*, it is at frame **100**). You can move it by clicking LMB  anywhere in the window, and you can even scroll forth and back the animation by clicking and dragging with this same mouse button. The actual frame number (or second value) is drawn near the pointer when you click or drag the time cursor – and obviously, it is always in the "current frame" numeric field of the [header](#).

Schlüsselbilder

Some keyframes of the active object (or active [Doc:2.5/Manual/Animation/Keyframes](#), etc.) are materialized by vertical colored lines at the frame they occur. I think only three different lpo types are drawn this way:

- **Gelb**

Non-action *Object* keys (location, rotation, etc.)

- **Orange**

Material keys (diffuse/specular/mirror colors, etc.)

- **dunkel Cyan**

keys forming [actions](#) (always pose of armature's bones, but also possibly objects' and shapes' lpos)

Note

Some lpo types aren't materialized at all (e.g. *Texture* or *Constraint* ones).

In [Timeline window](#), we have two "object" keyframes (frames **10** and **150**), one "material" keyframe (frame **40**), and no "action" keyframe.

Marker

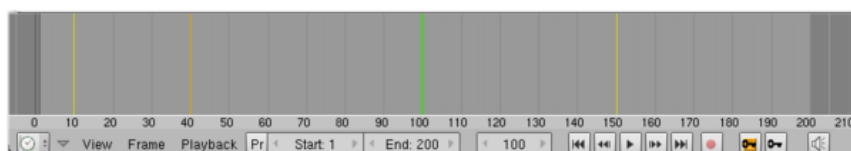
[Markers](#) werden als kleine Triangel dargestellt, mit ihren Namen in der Nähe.

Farb Codes sind::

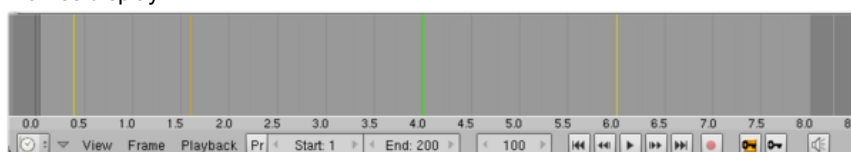
- **weiße Linie, schwarzer Text:** unausgewählte Marker
- **gelbe Linie, weißer Text:** ausgewählte Marker

Anpassen der Ansicht

Timeline window.






Frames display.



Seconds display.

This window is unidimensional – it only represents the time in your scene, along its horizontal axis. As frames are the fundamental unit of time in Blender, the timeline displays by default frame numbers, at its bottom.

The animation range is materialized by the lighter shade of gray (in *Timeline window*, from frame **1** to frame **200**).

This window behaves as any “area” in Blender: you can pan it with MMB  click-and-drag (left/right only, as this is an unidimensional window...), zoom in/out with Ctrl MMB  or Wheel , etc.

Ansichtsmenü

Dieses Menü kontrolliert was Sie sehen und wie Sie es sehen:

Maximize Window(CtrlArrowup)

This standard command makes this window full-screen. When it is maximized, this entry turns to *Tile Window*, to restore it to its previous size (CtrlArrowdown).

Lock Time to Other Windows

This is a cross-windows feature concerning all “time” windows (i.e. all windows which represent the time along their X axis). All these windows that have this option enabled will always show the same “time range”. When you modify it in one window (either by panning or scaling along its X axis), all the others are immediately updated.

Alles zeigen (Home)

Dieser Standard Befehl wird horizontal die Ansicht vergrößern die gesamte Animations Reichweite anzuzeigen (wie definiert vom *Start* und *End* Bild).

Center View(C)

This standard command will center the display on the current frame.

Jump to Prev Key (CtrlPageDown)

Jump to Next Key (CtrlPageUp)

These commands will make the cursor jump the the nearest previous or next keyframe of the active object.

Jump to Prev Marker (PageDown)

Jump to Next Marker (PageUp)

These commands will make the cursor jump the the nearest previous or next [marker](#).

Only Selected Data Keys

I think that when this is enabled, the *Timeline* should only draw the selected keyframes... But actually, it seems to always draw the keyframes of the active element!

Sekunden zeigen/Literal/ShowFrames (T)

By default, the time is materialized as frames, as it is internally in Blender. This menu entry enables you to rather show the time in seconds (based on the frames per second setting of the scene).

Very often storyboards are laid out in seconds. Choosing this display unit makes things a little easier than doing all that multiplying in your head.

Play Back Animation

Plays the animation from the current frame to end, and then cycling from start to end until you click the pause button (or hit Esc).

All windows matching the criteria set in the *Playback* menu (see below) will display the animation.

Bearbeiten

Frame Menü

Dieses Menü beinhaltet grundlegend die [Marker](#), mit zwei anderen nützlichen Optionen:

End-Frame setzen (E)

Start-Frame setzen (S)

Wie Sie vielleicht schon erraten können, setzen diese Optionen den momentanen Frame als Start-, oder End-Frame der Animation fest...

Wiedergabe

Wiedergabe Menü

This menu controls how the animation is played back, and where.

Continue Physics

The tool tip says: “During playback, continue physics simulations regardless of the frame number”. Don’t understand what it means...

Set Frames/Sec – This will pop-up a numeric field where you can specify a new fps setting.

Note: Remember the warning of the [introduction](#) about modifying this setting after having created some animations – it will speed up/down all existing ones...

Sync Playback to Frames/Sec

When enabled, it will force the playback to synchronize with the expected frame rate.

Note that when Blender has enough power to compute more frames per second than needed, it will stick to the specified frame rate. So this setting has only an effect when the animations are too heavy to be computed in real time: by default, Blender will

render all frames, effectively slowing down the playback. With this option enabled, it will drop (i.e. not compute them) as much frames as necessary to keep the normal playback rate.
See also the “sync audio” button of the header, [below](#).

The other options concerns what type of windows should be included in the playback initiated by this *Timeline* window. Obviously, the more windows are involved, the more CPU power you'll need...

Sequencer Fenster

When this option is set, all the *Video Sequence Editor* windows are included in the playback (whatever is there “display” mode: *Sequence* or one of the preview ones...).

See also the “sync audio” button of the header, [below](#).

Aufgenommene Bilder Fenster

When this option is set, all the *UV/Image Editor* windows are included in the playback.

Knöpfe Fenster

When this option is set, all the *Buttons Windows* are included in the playback (this allows you to see the evolutions of the values of the animated settings).

Animation Fenster

When this option is set, all the animation windows (i.e. the *Ipo Curve Editor*, *Action Editor* and *NLA Editor* ones) are included in the playback.

Alle 3D Windows

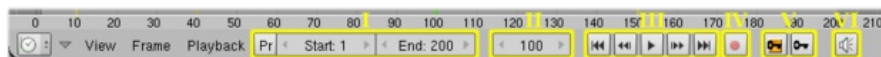
When this option is set, all the *3D Viewwindows* are included in the playback.

Top-Links 3D Fenster

When this option is set, only the top-left-most *3D Viewwindow* is included in the playback.

Kopfzeile Kontrollelemente

The header controls mostly mimic VTR ones:



The header of the *Timeline* window.

(I) The animation range

The first three controls concern the start and end frames of animation.

Start/End

The start and end frames! See also the *Set as Start/Set as End* entries of the [Frame menu](#) above.

Pr

Short for "Playback Range". This parameter is specified by *Start/End*, and are “linked” to the ones set in the *Anim* panel of the *Scene* context, *Render* sub-context (F10) – they are the same values. However, when you enable this *Pr* button, you can specify a new, temporary animation range, only available/effective for the realtime playbacks (initiated by the “play” button of the timeline, or the AltA shortcut). This is much useful when you have to work on a small piece of a big (long) animation!

(II) The current frame

The third numeric field displays, and allows you to modify, the current frame (as materialized by the green line cursor).

(III) VTR buttons

These five buttons allow you to navigate in your animation.

The center “play” button

Start the playback! When playing, it turns to a “pause” one, which... pauses (or stops) the playback.

The first and last buttons

respectively, send you to the start and end frame.

The other two

respectively send you to the previous/next shown keyframe.

(IV) Automation

The “record” red-dot button enables something often called as “automation”: it will add and/or replace existing keyframes of the active object when you transform it in a 3D view. Note that this work even during playback – the playback is just suspended during the transformations, and resumed once you have validated them.

When you enable this option, another drop-down *Auto-Keying Mode* list appears to its right, controlling the automation mode:

Replace Keys

This will only replace existing keyframes, never add new ones. Hence, your transformations will only have an effect when you do them at an already keyed frame.

Add/Replace Keys

This will replace existing keyframes, if any, or add new ones.

You'll find the same options (*Auto-Keying Enabled* button, which should be equivalent to the "record" one, and the *Auto-Keying Mode* drop-down list) in the *Edit Methods* tab of the *User Preferences* window. However, these ones seem to have no effect! But just below them, in the same *User Preferences* window, you have three toggle buttons that control which curves are automatically keyed by this tool:

Available

will add a key to all already existing lpo curves.

Needed

will add keys only when needed (i.e. only to lpo curves controlling properties that are changing).

Use Visual Keying

This is to be used with objects or bones that have certain constraints that can affect the key values. For example, setting a key on an object with a *Copy Location* constraint would normally set the key for it's unconstrained location. Enabling this option causes the key to be set for the constrained location.

Note that automation only works for transform properties (objects and bones), in the 3D views (i.e. you can't use it e.g. to animate the colors of a material in the *Buttons* window...).

(V) Inserting and deleting keys

The two "key" buttons allow you to insert (I) (orange background) or delete (AltI) the keys of the active object defined for the current frame.

This is not an easy-to-understand nor to-use feature! It tries to use the "context of the largest area" (an "area" is a Blender's window). E.g if your largest window is an *lpo Curve Editor* in *Material* "context", keys will be added to/removed from the active material lpo curves. If it's a 3D view, you'll get the same *Insert Key/Delete Key* menu as if you hit I/AltI in that 3D view...

(VI) Synchronize with the VSE sound

Enabling the "speaker" button has basically the same effect as enabling both the *Sync Playback to Frames/Sec* and *Sequencer Windows* options of the *Playback* menu. The playback now includes the sequencer, and uses its audio output as time-reference. Most useful during video editing...

Schlüsselbilder

Schlüsselbilder sind die Basis des Animierens. Schlüsselbilder definieren den Wert von Daten an einer bestimmten Bildposition.

Einfach gesagt, etwas ist "animiert" wenn sich etwas nach einiger Zeit verändert. In Blender bedeutet das Animieren von Objekten das Ändern von Eigenschaften wie der X-Position, oder den Rotkanalwert der Zerstreuungsfarbe von Material, und so weiter, während bestimmten Zeiten.

Wie erwähnt, ist Blenders fundamentale Einheit der Zeit das "Einzelbild", in englisch "Frame", welches für gewöhnlich nur den Bruchteil einer Sekunde, abhängig von der *Bildrate*, der Szene ausmacht.

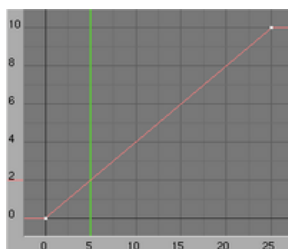
Da jede Animation mittels inkrementellen Änderungen über mehrere Bilder aufgebaut wird, werden diese Eigenschaften NICHT manuell für jedes Einzelbild geändert, weil:

- es würde Jahre dauern!
- es wäre sehr schwierig glatte Variationen der Eigenschaft zu erhalten (es sei denn Sie berechnen mathematische Funktionen und geben für jedes Einzelbild einen präzisen Wert an, was verückt wäre).

Das ist der Grund warum fast alle direkten Animationen mittels **Interpolation** bewerkstelligt werden.

Die Idee ist einfach: Sie legen einige "Kontrollpunkte" fest, "**Schlüsselbilder**" genannt, welche mehrere Bildpositionen auseinander liegen.

Zwischen diesen Schlüsselbildern, werden die Eigenschaftswerte von Blender berechnet (interpoliert) und eingefügt. Dadurch wird die Arbeitsbelastung des Animierers deutlich reduziert. Diese [Kurven](#) können im [Graphen Editorbearbeitet](#) werden.



Beispiel der Interpolation

Zum Beispiel, wenn Sie:

- einen Kontrollpunkt vom Wert **0** an der Bildposition **0**,
- einen Anderen vom Wert **10** an der Bildposition **25**,
- lineare Interpolation verwenden,

dann, bekommen wir an der Bildposition **5** einen Wert von **2**.

Das selbe gilt für alle Zwischenbilder: mit nur zwei Punkten, erhalten Sie ein Glättenwachstum von **0** bis **10** entlang der **25 Bilder**. Wenn Sie möchten, dass das Einzelbild **15** den Wert **9** bekommt, müssen Sie einen anderen Kontrollpunkt (oder ein Schlüsselbild) hinzufügen...

Schlüsselbilder erstellen


In der 3D Ansicht

Schlüsselbilder werden mit der Taste **I** angelegt. Ein Menü erscheint mit den folgenden Optionen:

- Position
- Rotation
- Skalieren
- PosRot
- Pos, Skalieren
- Pos, Rot, Skalieren
- Rot, Skal
- Visuelle Position
- Visuelle Rotation
- Visuelle Pos, Rot
- Delta-Position
- Delta Rotation
- Delta-Skalierung

In dem Eigenschaften Panel

Sie können Schlüsselbilder für fast alle veränderbaren Eigenschaften hinzufügen. Dass erreichen Sie durch das Überfahren der Maus über eine Eigenschaft und die Verwendung von Tasten:

- **I**. Das Feld wird Gelb, kennzeichnend dass ein Schlüsselbild gesetzt wurde.
- **Alt+i**. Löscht das aktuelle Schlüsselbild.
- **Alt+shift+i**. Löscht alle Schlüsselbilder auf diesem Kanal.
- **RMB**  auf eine Eigenschaft, und Schlüsselbild einfügen oder Einzelnes Schlüsselbild einfügen für Vektoreigenschaften.

In den Animations Editoren

Sie können Schlüsselbilder zu Kanälen, die über Schlüsselbilder verfügen, in mehreren Animations Editoren hinzufügen, welche auf anderen Seiten beschrieben sind::

- [Zeitstrahl](#)
- [Graphen Editor](#)
- [Dope Sheet](#)

Schlüsselbilder bearbeiten

Es existieren verschiedene Wege Schlüsselbilder zu bearbeiten, welche auf anderen Seiten oben aufgelistet behandelt werden.

Animationseditoren

Animation ist eine langwierige und schwierige Aufgabe. Deswegen gibt es viele Verschiedene Methoden die diesen Prozess beschleunigen oder den Arbeitsaufwand reduzieren können.

Für diese verschiedenen Methoden gibt es unterschiedliche Editorfenster:

Die [Zeitlinie](#)

Die Zeitlinie erlaubt keine Bearbeitung von Animationsdaten, kann aber Schlüsselbilder einer Szene darstellen.

Der [Graph Editor](#)

Ist die unterste Ebene des Animationsprozesses. Hier werden die Transformationen der Objekte als Kurven dargestellt bzw. können mit Kurven festgelegt werden.

Die [Dope Sheet](#)

Die Dope Sheet hat vier Modi die sich zwar alle ähneln aber mit anderen Animationsdaten arbeiten.

[Dope Sheet](#) Modus

Ein einfacher Schlüsselbildeditor der alle Schlüssel aller Kanäle in einer Szene darstellt.

[Aktions Editor](#) Modus

Ähnlich wie der Dope Sheet Modus, allerdings werden hier nur Aktionen für ein Objekt erstellt und bearbeitet.

[Formschlüssel Editor](#) Modus

Stellt Formschlüssel Animationen für ausgewählte Objekte dar.

[Stift](#) Modus

Stellt die Stift Daten dar.

Der [NLAeditor](#)

Die Idee hinter diesem höher leveligen Editor ist von nicht linearen Videoeditoren abgeleitet. Jedes Element repräsentiert eine Aktion die bewegt, dupliziert und verzerrt werden kann.



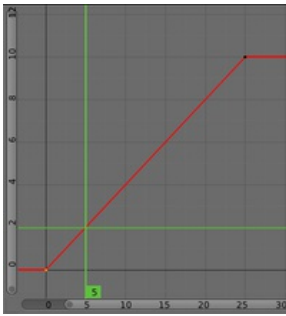
Es ist wichtig zu wissen wie die Animationsdaten aufgebaut sind!

- Schlüsselbilder formen Kurven
- Die in Aktionen zusammengefasst
- Im NLA Editor bearbeitet werden können.

F-Kurven

Wenn erstmal Animationsschlüssel für eine Aktion festgelegt sind können deren F-Kurven verändert werden.

Das Konzept der Interpolation



Beispiel einer Interpolation

Um etwas zu [animieren](#) ist müssen Veränderungen über eine Zeitspanne stattfinden. Das können alle möglichen Veränderungen sein, aber sie werden alle ähnlich animiert.

Es werden bestimmte Punkte mit Animationsschlüsseln festgelegt und damit nicht für alle Bilder einer Animation die Werte per Hand eingegeben werden müssen werden für alle zwischenliegenden Bilder Werte auf Basis von zwei Punkten und der dazwischen liegenden Kurve berechnet. Das nennt man Interpolation.

Zum Beispiel:

Bei einem Kontrollpunkt bei Bild 0 mit dem Wert 0 und einem anderen mit Wert 10 bei Bild 25 und einer linearen Interpolation ergibt sich ein Wert von 2 bei Bild 5.

Einstellungen

F-Kurven haben drei zusätzliche Eigenschaften die die Interpolation zwischen Punkten, den Hebelyp und das Verhalten der Kurve beeinflussen.

Interpolations Modi

Es gibt drei Interpolationsmodi für den Graphen zwischen den Animationsschlüsseln:

Konstante

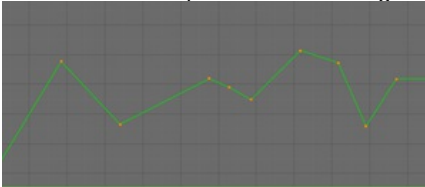
Keine Interpolation. Der Wert bleibt gleich dem beim letzten Animationsschlüssel.



Konstante.

Linear

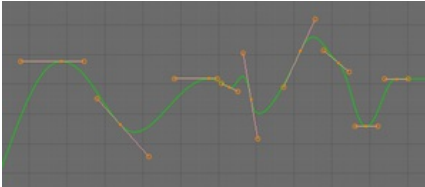
Diese einfach Interpolation bildet eine gerade Linie zwischen den Werten des letzten und nächsten Animationsschlüssels.



Linear.

Bezier

Diese Standardmethode verbindet die Animationsschlüssel mit Bezierkurven und erzeugt glatte Animationen.



Bézier.

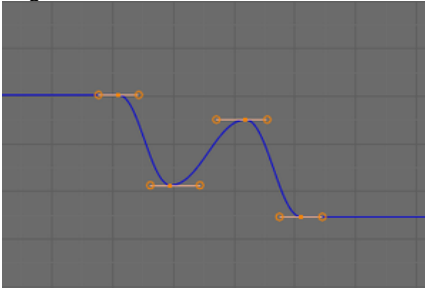
Extrapolation

(⇧ ShiftE, or Channel » Extrapolation Mode)

Extrapolation bestimmt das Verhalten der Kurven vor dem ersten und nach dem letzten Animationsschlüssel. Es gibt zwei Extrapolationsmodi:

Konstant

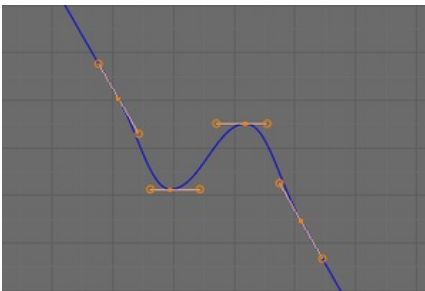
Fügt konstante Funktionen ein.



konstante Extrapolation

Linear

Kurven werden mit linearen Funktionen fortgesetzt.



Linearextrapolation

Zusätzliche Extrapolationswerkzeuge sind unter [F-Curve Modifikatoren](#) zu finden.

Hebel Typen

Für Bezierkurven gibt es zusätzlich noch Einstellungen für die Hebel. Der Typ des Hebels kann mit V geändert werden.

Automatik

Animationsschlüssel werden automatisch interpoliert.



Auto Hebel

Vektor

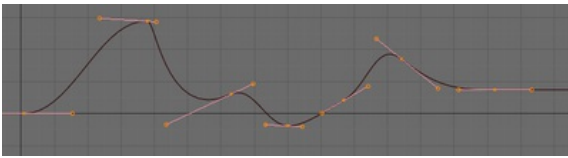
Erzeugt lineare Hebel an Animationsschlüsseln. Wenn der Hebel bewegt wird, wird er zu einem freien Hebel.



Vektor Hebel

Ausgerichtet

Hebel behalten ihre Ausrichtung wenn sie bewegt werden und ihre Kurventangente wird beibehalten.



Ausgerichtete Hebel

Freie

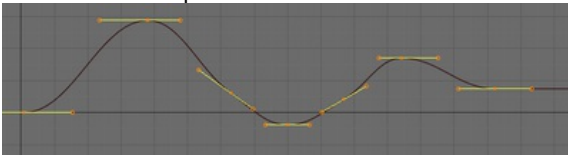
Der Hebel kann frei bewegt werden.



Freie Hebel

Auto Klammer

Auto handles clamped to not overshoot

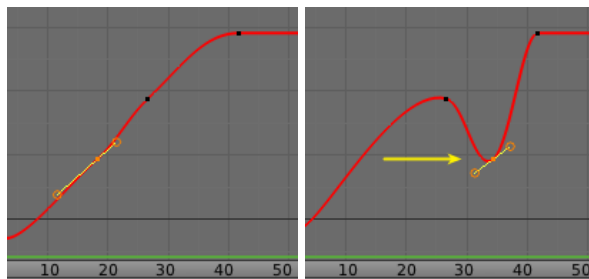


Auto clamped handles

Richtung der Zeit

Obwohl F-Kurven Bezierkurven sehr ähnlich sind, gibt es einige wichtige Unterschiede. F-Kurven sind mehr Funktionen im mathematischen Sinn, als Kurven. Es können nicht zwei Wertpunkte mit unterschiedlichen Werten der selben Eigenschaft an einem Zeitpunkt existieren. Blender ordnet also selbstständig alle Animationsschlüssel in eine Reihenfolge. Wird ein Punkt so vertauscht das die Reihenfolge zeitlich nicht stimmen würde, wird die Reihenfolge geändert, sodass sie stimmt.

Two control points switching: the curve can't go back in time!



Vor dem Verschieben in der normalen Reihenfolge

Nach dem Verschieben in der neuen Reihenfolge

F-Kurve

Der F-Curve modifiers are similar to object modifiers, in that they add non-destructive effects, that can be adjusted at any time, and layered to create more complex effects.

Hinzufügen des Modifikators

Das Panel des F-Kurve Modifikators befindet sich im Eigenschaften-Panel. Select a curve by selecting one of its curve points, or by selecting the channel list. Click on the **Add Modifier** button and select a modifier.

To add spin to an object or group, select the object/group and add a keyframe to the axis of rotation (x,y, or z)

Go to the Graph Editor.....make sure the f-curves properties panel is visible (View > Properties)

>Add Modifier > Generator

Types of Modifiers

Generator

Generator creates a Factorized or Expanded Polynomial function. These are basic mathematical formulas that represent lines, parabolas, and other more complex curves, depending on the values used.

Additive

This option causes the modifier to be added to the curve, instead of replacing it by default.

Poly Order

Specify the order of the polynomial, or the highest power of 'x' for this polynomial. (number of coefficients - 1).

Change the Coefficient values to change the shape of the curve. See [\[The Wikipedia Page\]](#) for more information on polynomials.

Eingebaute Funktionen

Unten stehen zusätzliche Formulare, jede mit den selben Option zur Formkontrolle. Erkundigen Sie sich bei mathematischen Referenzen für detaillierte Informationen für jede einzelne Funktion.

- Sinus
- Kosinus
- Tangens
- Square Root
- Natural Logarithm
- Normalized Sine ($\sin(x)/x$)

Amplitude

Adjusts the Y scaling

Phase Multiplier

Adjusts the X scaling

Phase Offset

Adjusts the X offset

Value Offset

Adjusts the Y offset

Envelope

Allows you to adjust the overall shape of a curve with control points.

Reference Value

Set the Y value the envelope is centered around.

Min

Lower distance from Reference Value for 1:1 default influence.

Max

Upper distance from Reference Value for 1:1 default influence.

Add Point

Add a set of control points. They will be created at the current frame.

Fra:

Set the frame number for the control point.

Min

Specifies the lower control point's position.

Max

specifies the upper control point's position.

Cycles

Cycles allows you add cyclic motion to a curve that has 2 or more control points. The options can be set for before and after the curve.

Cycle Mode

Repeat Motion

Repeats the curve data, while maintaining their values each cycle.

Repeat with Offset

Repeats the curve data, but offsets the value of the first point to the value of the last point each cycle.

Repeat Mirrored

Each cycle the curve data is flipped across the X-axis.

Before/After Cycles

Set the number of times to cycle the data. A value of 0 cycles the data infinitely.

Noise

Modifies the curve with a noise formula. This is useful for creating subtle or extreme randomness to animated movements, like camera shake.

Blend Type

Replace

Adds a -.5 to .5 range noise function to the curve.

Add

Adds a 0 to 1 range noise function to the curve.

Subtract

Subtracts a 0 to 1 range noise function to the curve.

Multiply

Multiplies a 0 to 1 range noise function to the curve.

Scale

Adjust the overall size of the noise. Values further from 0 give less frequent noise.

Strength

Adjusts the Y scaling of the noise function.

Phase

Adjusts the random seed of the noise.

Depth

Adjusts how detailed the noise function is.

Python

Limits

Limit curve values to specified X and Y ranges.

Minimum/Maximum X

Cuts a curve off at these frames ranges, and sets their minimum value at those points.

Minimum/Maximum Y

Truncates the curve values to a range.

Stepped

Gives the curve a stepped appearance by rounding values down within a certain range of frames.

Step Size

Specify the number of frames to hold each frame

Offset

Reference number of frames before frames get held. Use to get hold for '1-3' vs '5-7' holding patterns.

Use Start Frame

Restrict modifier to only act before its 'end' frame

Use End Frame

Restrict modifier to only act after its 'start' frame

Einleitung zur Physik Simulation

Dieses Kapitel deckt verschiedene erweiterte Blender Effekte ab, die oft zum Simulieren echter physikalischer Phänomene genutzt werden, wie:

- Rauch
- Regen
- Staub
- Kleidung
- Wasser
- Wackelpudding

[Partikel Systeme](#) können verwendet werden um einige Dinge zu simulieren: Haare, Gras, Rauch, Flocken.

[Haare](#) sind Teil des Partikel-Systems, und können für faserähnliche Objekte, wie Haar, Pelz, Gras, Federn usw. verwendet werden.

[Weiche Körper](#) sind für alles nützlich, was dazu tendiert, sich in Reaktion von Kräften wie Gravitation oder Wind, oder beim Kollidieren mit anderen Objekten, zu verbiegen oder zu defomieren ... Es kann für Haut, Gummi und sogar für Kleidung verwendet werden, auch wenn es seperate, für stoffähnliche Objekte spezifische [Kleidungssimulation](#) gibt.

[Starre Körper](#) können dynamische Gegenstände vortäuschen, die ziemlich starr sind.

[Flüssigkeiten](#), die Flüssigkeiten und Gase umfassen, können einschließlich [Rauchs](#) simuliert werden.

[Kraftfelder](#) können das Verhalten von Simulationen modifizieren.

Gravitation

Gravitation ist eine globale Einstellung, die dasselbe auf alle Physik-Systeme in einer Szene anwendet, die im Szene-Reiter gefunden werden können. Dieser Wert hat allgemein einen Vorgabewert, von -9.810 in der Z Achse, welche die Anziehungskraft in der echten Welt ist. Das Senken dieses Wertes würde eine niedrigere oder höhere Anziehungskraft simulieren.

Bemerken Sie, dass Sie den Gravitationswert pro Physik-System im Field Weights Reiter herunterregeln können.

Haare

Sobald auf den Haar Modus gesetzt, erzeugt das Partikelsystem nur noch statische Partikel, die für Haare, Fell und Gras verwendet werden können.

Wachsen lassen

Der erste Schritt um Haare zu erstellen ist, die Anzahl der einzelnen Haare und deren Länge vorzugeben.

Der komplette Weg der Partikel wird danach berechnet. Das heißt, dass ein Haar alles tut, was ein Partikel auch tun würde. Ein Haar ist so lang, wie der Weg, den ein Partikel in einer Zeit von 100 Frames zurücklegen würde. Anstatt jeden Frame des Haars, Punkt für Punkt, zu rendern werden Kontrollpunkte mit einer Interpolierung, den Segmenten, berechnet.

Styling

Der nächste Schritt ist die Haare zu stylen. Das einfache Aussehen der Haare lässt sich ändern indem man die [Physics Settings](#) ändert.

Ein erweiterter Weg das Aussehen der Haare zu veränderst ist [Children](#) zu benutzen. Dadurch werden zusätzliche Haare auf Basis der Originalen und einiger Einstellungen hinzugefügt.

Haare können auch interaktiv im [Particle Mode](#) bearbeitet werden. In diesem Modus können Haare gekämmt, geschnitten, verlängert, etc. werden. Dabei werden die Partikel Einstellungen deaktiviert.

Animieren

Die Haare können dynamisch werden, indem man die Kleidungssimulation verwendet. Dies ist in [Hair Dynamics](#) beschrieben.

Rendern

Blender can render hairs in several different ways. Materials have a Strand section, which is covered in the materials section in the [Strands Page](#).

Hair can also be used as a basis for the [Particle Instance modifier](#), which allows you to have a mesh be deformed along the curves, which is useful for thicker strands, or things like grass, or feathers, which may have a more specific look.

Optionen

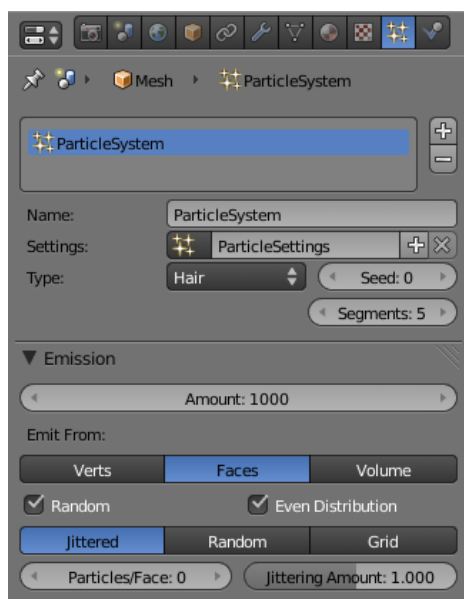


Image 4a: Settings for a Hair particle system.

Regrow

Lässt die Haare jeden Frame nachwachsen.

Advanced

Aktiviert die selben Einstellungen wie im Emitter Modus.

Emission

Amount

Anzahl der einzelnen Haare. Verwende so wenig Partikel wie möglich, besonders, wenn du eine Animation planst. Allerdings werden genug benötigt, um genug Kontrolle über die Haare zu haben. Für einen normalen Haarschnitt sind 1000-2000 ausreichend. Um einen Körper mit Fell zu überziehen werden natürlich mehr benötigt. Damit die Haare nicht dünn erscheinen werden später Children verwendet.

Haar Dynamik

Einstellungen um Bewegung in die Haare zu bekommen. Siehe [Hair Dynamics](#).

Darstellung

Rendered

Haare werden als Kurven geseichnet.

Path

Es werden nur die Endpunkte der Haare gezeichnet.

Steps

Die Zahl der Segmente (Kontrollpunkte minus 1) des Haares. zwischen Kontrollpunkt und Segment werden die Haare interpoliert. Die Zahl der Kontrollpunkte ist wichtig:

- für Animationen, da Kontrollpunkte wie Vertecies behandelt werden. Eine höhere Kontrollpunkanzahl bedeutet längere Berechnungszeiten.
- für interaktives Beatbeiten, da nur die Kontrollpunkte bewegt werden können (die Zahl der Kontrollpunkte kann im Particle Modus neu berechnet werden).

10 Segmente sollten auch für sehr langes Haar ausreichend sein, 5 Segmente für kürzeres Haar, und 2 oder 3 sollten genug für kurzes Fell sein.

Children

See [Children](#).

Render

Hair can be rendered as a Path, Object, or Group. See [Particle Visualization](#) for descriptions.

Benutzung



Image 4b: Particle systems may get hairy...

- [Fur Tutorial](#), which produced (*Image 4b*). It deals especially with short hair.

Motion Tracking

Einführung

Motion tracking ist eine neu verfügbare Technik in Blender. Sie befindet sich noch in der Entwicklung, unterstützt aber schon Einfache Operationen für 2D motion tracking, 3D motion tracking, und camera solution. Sie wurde schon in Produktionen wie "Tears of Steel" eingesetzt.

Einstieg

Motion Tracking ist in der aktuellen Version verfügbar und direkt einsetzbar.

Hier ist eine kurze Beschreibung von Motion-Tracking Tools in Blender

Supervised 2D tracking

Es gibt keinen gemeinsamen Algorithmus, der für alle Arten von Filmmaterial, Feature-Punkte und ihre Bewegungen verwendet werden kann. Solche Algorithmen können erstellt werden, aber diese wären sehr langsam und nicht fehlerfrei, deshalb ist die einzige Möglichkeit für 2D Tracking, manuell einen Algorithmus und dessen Einstellungen auszuwählen. Die momentanen Standardeinstellungen funktionieren für einen normalen, nicht stark verwackelten schon ziemlich gut.

Verbessertes 2D Tracking steht schon in unserer TODO-Liste, hat jedoch im Moment wenig Priorität. Wenn Sie nicht sicher sind welcher Algorithmus der Richtige ist und sie diesen Artikel nicht lesen wollen, dann spielen sie mit den Einstellungen, bis sie eine gefunden haben, die funktioniert

Manual lens calibration using grease pencil and/or grid

All cameras record distorted videos. Nothing can be done about this because of the manner in which optical lenses work. For accurate camera motion, the exact value of the focal length and the "strength" of distortion are needed.

Currently, focal length can be automatically obtained only from the camera's settings or from the EXIF information -- there are no tools inside Blender which can estimate it. But there are some tools which can help to find approximate values to compensate for distortion. There are also fully manual tools where you can use grid which is getting affected by distortion model and deformed cells defines straight lines in the footage. You can also use grease pencil for this - just draw line which should be straight on the footage using poly line brush and adjust distortion values to make grease pencil match lines on the footage.

To calibrate your camera more accurately, use the grid calibration tool from OpenCV. OpenCV is using the same distortion model, so it shouldn't be a problem.

Camera motion solving

Despite that there's no difference in solving camera motion and object motion from math's point of view, only camera solving is currently supported. And it still has some limitations, like unsupported solve of tripod motions or dominant plane motions (where all track-able features belong to one plane). These limitations are planned to be solved in the future.

Basic tools for scene orientation and stabilization

After solve, something is needed to orient real scene in 3D scene for more convenient compositing. There's tools to define floor, scene origin, and X/Y axis to perform scene orientation.

If something is needed to stabilize video from camera to make the final result looks nicer, 2D stabilization is available to help. It stabilizes video from camera which can compensate camera jumps and tilt.

Basic nodes for compositing scene into real footage

Some new nodes were added to compositor to composite scene into footage in easier way. So, there are nodes for 2D stabilization, distortion and undistortion which are easy to use.

Not implemented tools

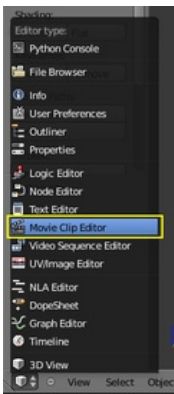
Some tools aren't available in Blender yet, but they are in our TODO list. So there's currently no support for such things as rolling shutter filtering, object motion solving, motion capturing. But you can try to hack this stuff using currently implemented things.

Manual

Movie Clip Editor

Almost all motion tracking tools are concentrated in editor called Movie Clip Editor. Currently it doesn't have any tools which aren't related on motion tracking, but in future it can be expanded to be used for masking and so, that's why it's called in this more abstract from motion tracking manner.

This editor can be found in list of editor types.



Editor type menu

When you'll switch to Movie Clip Editor interface would change in the following way.



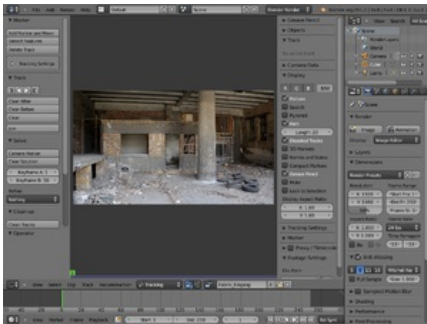
Movie Clip Editor interface

Next thing which is logical to do is to open new video clip to start working with. There are several ways to this:

- Use **Open** button from movie editor header
- Use Clip » Open menu
- Use AltO> shortcut

Both of movie files and image sequences can be used in the clip editor. If you're using image sequence there's one limitation on naming of files: last group of numbers should be increasing continuously.

So, when movie clip is loaded into clip editor, extra panels are displayed in the interface.



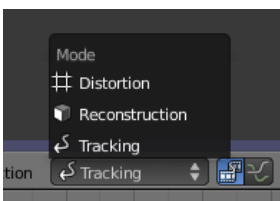
Movie Clip Editor with opened clip

There are plenty of new tools appeared on the screen and here's short description of all of them.

First of all, it should be mentioned, that camera solver consists of three quite separated steps:

- 2D tracking of footage
- Camera intrinsics (focal length, distortion coefficients) specification/estimation/calibration
- Solving camera, scene orientation, scene reconstruction

Tools in clip editor are split depending on step they're used at, so interface isn't messed up with scene orientation tools when only 2D tracking can be done. Currently displayed tools category can be changed using Mode menu which is in editor header.



Movie Clip Editor mode menu



But almost all operators can be called from menus, so it's not necessary to change mode every time you want to use a tool which is


associated with a different editor mode.

In tracking mode only tools which are related on tracking and camera solving are displayed. Camera solving tools are exposed here because it's after solving you'll most probably want to re-track existing tracks or place new tracks to make solving more accurate.

Tools available in tracking mode

Markers panel

- **Add Marker and Move** operator places new marker at the position of mouse (which is under the button in this case, not ideal but it's just how things work) and then it can be moved to needed location. When it's moved to needed position, LMB  can be used to finish placing new marker. Also,  Enter and Space can be used to finish placing marker.

But it's faster to use Ctrl LMB  to place markers directly on the footage. This shortcut will place marker on place you've clicked. One more feature here: until you've released mouse button you can adjust marker position moving mouse and using track preview widget to control how accurate marker is placed.

- **Detect Features** operator detects all possible to features on current frame and places markers at this features. This operator doesn't take in account other frames, so it can place markers on features which belongs to moving object and if camera is turning away from this shot, no markers would be placed on frames after camera moved away.

There are several properties for this operator:

Placement is used to control where to place marker. By default, they'll be added on the whole frame, but you can also outline some interesting for detecting areas with grease pencil and place markers only inside outlined area. That's how "Inside Grease Pencil" placement variant works. You can also outline absolutely not interesting areas (like trees, humans and so) and place markers outside of this areas. That's how "Outside Grease Pencil" placement variant works.

Margin controls distance from image boundary for created markers. If markers would be placed too close to image boundary, they'll fail to track really soon and they should be deleted manually. To reduce amount of manual cleaning up this parameter can be used.

Trackability limits minimal trackability for placing markers. This value came from feature detection algorithm and basically it means: low values means most probably this feature would fail to track very soon, high value means it's not much such track. Amount of markers to be added can be controlled with this value.

Distance defines minimal distance between placed markers. it's needed to prevent markers placed too close to each other (such placement can confuse camera solver).

- **Delete Track** is quite self-explaining operator which deletes all selected tracks.

Track panel

- First row of buttons is used to perform tracking of selected tracks. Tracking can happen (in order of buttons):
 - Backward one frame
 - Backward along the sequence
 - Forward along the whole sequence
 - Forward one frame

This operator depends on settings from Tracking Settings panel which would be described later.

If during sequence tracking algorithm returned failure of tracking some markers, they'll be disabled and tracking will continue for rest of markers. If algorithm returns failure when tracking frame-by-frame, marker wouldn't be disabled and most possible position of feature on new frame would be used.

- **Clear After** deletes all tracked and keyframed markers after current frame for all selected tracks.
- **Clear Before** deletes all tracked and keyframed markers before current frame for all selected tracks.
- **Clear** clears all markers except current from all selected tracks.
- **Join** operator joins all selected tracks into one. Selected tracks shouldn't have common tracked or keyframed markers at the same frame.

Solve panel

Camera Motion operator solves motion of camera using all tracks placed on the footage and two keyframes specified on this panel. There are some requirements:

- There should be at least 8 common tracks on the both of keyframes
- It should be well noticeably parallax effect between this two keyframes

If everything came smooth during solve, average reprojection error would be reported to the information space and to clip editor header. Reprojection error means average distance between reconstructed 3D position of tracks projected back to footage and original position of tracks. Basically, reprojection error below 0.3 means accurate reprojection, 0.3-3.0 means quite nice solving which still can be used. Values above 3 means some tracks should be tracked more accurate or that values for focal length or distortion coefficients were set incorrectly.

Refine option specifies which parameters should be refined during solve. Such kind of refining is useful when you aren't sure about some camera intrinsics and solver would try to find best parameter for that intrinsics. But you still have to know approximate initial values - it'll fail to find correct values if they were set completely incorrect initially.

Cleanup Panel

This panel contains single operator and it's settings. This operator cleans up bad tracks: tracks which aren't tracked long enough or which failed to reconstruct accurate. Threshold values can be specified from slides below the button. Also, several actions can be performed for bad tracks:

- They can be simply selected
- Bad segments of tracked sequence can be removed
- The whole tracks can be deleted

Clip Panel

This panel currently contains the single operator **Set as background** which sets currently editing clip as camera background for all visible 3D viewports. If there's no visible 3D viewports or clip editor is opened in full screen, nothing will happen.

Properties available in tracking mode

Grease Pencil Panel

It's a standard grease pencil panel where new grease pencil layers and frames can be controlled. There's one difference in behavior of grease pencil from other areas - when new layer is created "by-demand" (when making stroke without adding layer before this) default color for layer would be set to pink. It makes stroke easy to notice on all kinds of movies.

Objects Panel



Objects Panel in clip editor

This panel contains list of all objects which can be used for tracking, camera or object solving. By default there's only one object in this list which is used for camera solving. It can't be deleted and other objects can't be used for camera solving, all added objects are used for object tracking and solving only. These objects can be referenced from Follow Track and Object Solver constraints. Follow Track uses camera object by default.

New objects can be added using **+** and active object can be deleted by **-** button. Text entry at the bottom of this panel is used to rename active object.

If some tracks were added and tracked to wrong object, they can be copied to another object using Track » Copy Tracks and Track » Paste Tracks.

Usage for all kind of objects (used for camera and object tracking) is the same: track features, set camera data, solve motion. Camera data is sharing between all objects and refining of camera intrinsics happens when solving camera motion only.

Track Panel



Track Panel in clip editor

First of all, track name can be changed in this panel. Track names are used for linking tracking data to other areas like follow track constraint.

Next thing which can be controlled here is marker's enabled flag (using button with eye icon). If marker is disabled, it's position wouldn't be used neither by solver nor by constraints.

Button with lock at the right of button with eye means track is locked. Locked tracks means it can't be edited at all. This helps to prevent accidental changes to tracks which are "finished" (tracked accurate along the whole footage).

Next widget placed on this panel is called "Track Preview" and it displays content of pattern area. This helps to check how good tracking is happening (control there's no slides from original position) and also helps to move track back to needed position. Moving of track can happen directly from this widget by mouse slide.

If anchor is used (position on image which is tracking is different from position which is used for parenting) preview widget will display

area around anchor position. Such configuration helps for masking some things when there's no good feature at position where mask corner should be placed. Details of this technique would be written later.

There's small area below the preview widget which can be used to enlarge vertical size of preview widget (area is highlighted by two horizontal lines).

Next setting is channels control. Tracking happens in grayscale space, so a high contrast between feature and background yields more accurate tracking. In such cases disabling some color channels can help.

The last thing is custom color and preset for it. This setting overrides default marker color used in clip editor and 3D viewport and it helps to distinguish different type of features (for example features on far plane or near plane and so). Color is also can be used for "grouping" tracks so the whole group of tracks can be selected by color using Select Grouped operator.

- **Marker Tip 1:** To use good points for tracking, use points in the middle of the footage timeline and track backwards and forwards from there. This will provide greater chance of the marker and point staying in the camera shot.

Camera Data Panel

This panel contains all settings of camera used for screening movie which is currently editing in clip editor.

First of all, predefined settings can be used here. New presets can be added or unused presets can be deleted. But such settings as distortion coefficients and principal point aren't included into presets and should be filled in even if camera presets were used.

- **Focal Length** is already self-described, it's a focal length with which movie was shoot. It can be set in millimeters and pixels. In most cases focal length is given in millimeters, but sometimes (for example in some tutorials in the Internet) it's given in pixels. In such cases it's possible to set it directly in units it's known in.
- **Sensor Width** is a width of CCD sensor in camera. This value can be found in camera specifications.
- **Pixel Aspect Ratio** is a pixel aspect of CCD sensor. This value can also be known from camera specifications, but can also be guessed. For example, you know that footage should be 1920x1080 but images itself are 1280x1080. In this case pixel aspect is:

$$1920 / 1280 = 1.5$$

- **Optical Center** is an optical center of lens used in camera. In most cases it's equals to image center, but it can be differ in some special cases. Check camera/lens specifications in such cases. To set optical center to center of image, there's **Center** button below sliders.
- **Undistortion K1, K2 and K3** is a coefficients used to compensate lens distortion happened on movie shooting. Currently this values can be tweaked by hand only (there's no calibration tools yet) using tools available in Distortion mode. Basically, just tweak K1 until solving would be most accurate for known focal length (but also take grid and grease pencil into account to prevent "impossible" distortion)

Display Panel

This panel contains all settings which controls things displayed in clip editor.

- **R, G, B** and **B/W** buttons at the top of this panel are used to control color channels used for frame preview and to make the whole frame grayscale. It's needed because tracking algorithm works with grayscale image and it's not always obvious to see which channels disabled will increase contrast of feature point and reduce noise.
- **Pattern** can be used to disable displaying of rectangles which are correspond to pattern areas of tracks. In some cases it helps to make clip view more clear and check how good tracking is.
- **Search** can be used to disable displaying of rectangles which are correspond to search areas of tracks. In some cases it helps to make clip view more clear and check how good tracking is. Search areas for selected tracks only would be displayed.
- **Pyramid** makes the highest pyramid level be visible. What pyramid itself is would be described later in Tracking Settings panel section, but basically it helps to determine how much track is allowed to move from one frame to another.
- **Track Path** and **Length** controls displaying of path of tracks. So way in which tracks are moving can be visible looking at the only one frame. It helps to determine if track jumps from it's position or not.
- **Disabled Tracks** makes possible to hide all tracks which are disabled on current frame. This helps to make view more clear to control if tracking happens accurate enough.
- **Bundles** makes sense after solving movie clip and it works in the following way: solved position of each track gets projected back to movie clip and displayed as small point. Color of point depends on distance of projected coordinate and original coordinate: if they are close enough point would be green, otherwise it'll be red. This helps to find tracks which weren't solved nicely and need to be tweaked.
- **Track Names and Status** displays such information as track name and status of track (if it's keyframed, disabled, tracked or estimated). Names and status for selected tracks is displayed.
- **Compact Markers**. Way in which markers are displayed (black outline and yellow foreground color) makes tracks be visible on all kind of footages (both dark and light). But sometimes it can be annoying and this option will make displaying of marker in more compact manner - outline would be replaced by dashed black lines drawing on top of foreground, so markers areas would be 1px thick only.
- **Grease pencil** means if grease pencil strokes are allowed to be displayed and made.
- **Mute** changes displaying on movie frame itself with black square, it helps to find tracks which are tracked inaccurate or which weren't tracked at all.
- **Grid** (available in distortion mode only) displays grid which is originally orthographic but was affected by distortion model. This grid can be used for manual calibration - distorted lines of grids are equal to straight lines in the footage.
- **Manual Calibration** (available in distortion mode only) applies distortion model for grease pencil strokes. This option also helps to perform manual calibration. More detailed description of this process would be made later.
- **Stable** (available in reconstruction mode only). This option makes displaying frame be affected by 2D stabilization settings. It's only preview option which doesn't actually changes footage itself.

- **Lock to Selection** makes editor be displaying selected tracks on the same screen position along the whole footage during playback or tracking. This option helps to control tracking process and stop it when track is starting sliding off or when it jumped.
- **Display Aspect Ratio** changes aspect ratio for displaying only. It does not affect on tracking or solving process.

Tracking Settings Panel

Common options

This panel contains all settings for 2D tracking algorithms. Depending on which algorithm is used, different settings would be displayed, but there are also few common for all trackers settings:

Adjust Frames controls which patterns are getting tracking, if be more precise, pattern from which frame is getting tracked. Here's an example which should make things more clear.

Tracker algorithm receives two images inside search area and position of point to be tracked in first image. And tracker tries to find position of point from first image on second image.

Now, how tracking of sequence happens. Second image is always image created from frame at which position of marker isn't known (next tracking frame). But different first image can be send to tracker. Most commonly used combinations:

- Image created from frame on which track was keyframed. This configuration prevents sliding from original position (because position which corresponds most to the original pattern is returning by tracker), but it can lead to small jumps and can lead to failures when feature point is getting deformed due to camera motion (perspective transformation, for example). Such configuration is used if **Adjust Frames** is set to 0.
- Image created from current frame is sending as first image to the tracker. In this configuration pattern is tracking between two neighbor frames and it allows to deal with cases of high transformations of feature point but can lead to sliding from original position, so it should be controlled. Such configuration is used if **Adjust Frames** is set to 1.

If **Adjust Frames** is greater than 1, behavior of tracker would be like keyframes for tracks are creating every **Adjust Frames** frames and tracking between keyframed image and next image is used.

Speed can be used to control speed of sequence tracking. This option makes nothing with quality of tracking it's just helps to control if tracking happens accurate. In most of cases tracking happens much faster than realtime playing and it's difficult to notice when track began to slide off from position. In such cases **Speed** can be set to Double or Half to add some delay between tracking two frames, so slide off would be noticed earlier and tracking process can be cancelled to adjust positions of tracks.

Frames Limit controls how many frames can be tracked when Track Sequence operator is called. So, each Track Sequence operation would track maximum **Frames Limit** frames. This also helps to notice slide off of tracks and correct them.

Margin can be used to make tracks disabled when they are becoming too close to image boundary. This slider controls "too close" in pixels.

KLT tracker options

KLT tracker is an algorithm used by default. It allows to track most kinds of feature points and it's motion. It's using pyramid tracking which works in the following way. Algorithm tracks larger image than pattern first to find general direction of motion. Then it tracks a bit smaller image to refine position from first step and make final position more accurate. This continues several times. Number of steps of such tracking is equal to **Pyramid Level** option and we tell that on first step tracking happens for highest pyramid level. So Pyramid Level=1 is equal to pattern itself, and each next level doubles tracking image by 2.

Search area should be larger than highest pyramid level and "free space" between search area and highest pyramid level defines how much feature can be moved from one frame to another.

Default settings should work in most of general cases, but sometimes pyramid level should be changed. For example, when footage is blurry, adding extra pyramid levels helps to track them.

This algorithm can fail in situations when feature point is moving in one direction and texture around feature point is moving in another direction.

SAD tracker options

On each step SAD tracker reviews the whole search area and finds pattern on second image which is mostly like pattern which is getting tracking. This works pretty fast, but can fail in several cases. For example, when there're another feature point which looks like tracking feature point appears in search area. In this case SAD will lead to jump of track from one feature to another.

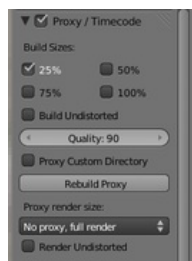
Correlation defines threshold value for correlation between two patters which is still considering as successful tracking. 0 means there's no correlation at all, 1 means correlation is full.

There's one limitation: currently, it works for features 16x16 only.

Marker Panel

This panel contains numerical settings for marker position, pattern and search areas dimensions, offset of anchor point from pattern center. All sliders are self-explainable.

Proxy / Timecode Panel



Proxy / Timecode
Panel in clip editor

This panel contains options used for image proxies and timecodes for movies.

Proxy allows to display images with lower resolution in clip editor. This can be helpful in cases when tracking of 4K footage is happening on machine with small amount of RAM.

First four options are used to define which resolutions of proxy images should be built. Currently it's possible to build images 25%, 50%, 75% and 100% from original image size. Proxy size of 100% can be used for movies which contains broken frames which can't be decoded.

Build Undistorted means that proxy builder would also create images from undistorted original images for sizes set above. This helps to have faster playback of undistorted footage.

Generated proxy images are encoding using JPEG and quality of JPEG codec is controlled by **Quality** slider.

By default, all generated proxy images are storing to <path of original footage>/BL_proxy/<clip name> folder, but this location can be set by hand using **Proxy Custom Directory** option.

Rebuild Proxy will regenerate proxy images for all sizes set above and regenerate all timecodes which can be used later.

Use Timecode Index can (and better be used) for movie files. Basically, timecode makes frame search faster and more accurate. Depending on your camera and codec different timecodes can make better result.

Proxy Render Size defines which proxy image resolution is used for display. If **Render Undistorted** is set, then images created from undistorted frame would be used. If there's no generated proxies, render size is set to "No proxy, full render" and render undistorted is enabled, undistortion will happen automatically on frame draw.

Tools available in reconstruction mode



Proxy / 2D
Stabilization Panel
in clip editor

There's one extra panel which is available in reconstruction mode - 2D stabilization panel.

This panel is used to define data used for 2D stabilization of shot. Several options are available in this panel.

First of all, it's list of tracks used to compensate camera jumps, or location. It works in the following way: it gets tracks from list of tracks used for location stabilization and finds median point of all these tracks on first frame. On each other frame algorithm makes this point have the same position in screen coordinates by moving the whole frame. In some cases it's not needed to fully compensate camera jumps and **Location Influence** can be used in such cases.

Camera can also rotate a bit adding some tilt to the footage. There's **Stabilize Rotation** option to compensate this tilt. Extra single track is needed to set for this and it works in the following way. On first frame of movie, this track is getting connected with median point of tracks from list above and angle between horizon and this segment is keeping constant along the whole footage. Amount of rotation applying on the footage can be controlled by **Rotation Influence**.

If camera jumps a lot, it'll be noticeable black areas near image boundaries. To get rid of these black holes, there's **Autoscale** option which finds smallest scale, applying which on the footage all black holes near image boundaries would eliminate. There's option to control maximal scale factor (**Maximal Scale**) and amount of scale applying on the footage (**Scale Influence**)

Render Engines

- Blender Internal
- [Cycles](#)

Rendering

Rendering ist der finale Prozess, in der ein 2D Bild entsprechend der erstellten 3D-Szene erzeugt wird. Sie können ein Bild auf ihrem Computer rendern, oder, da es ein rechenintensiver Prozess ist, in einer Render Farm, welche ein Netzwerk von PC's ist, die an verschiedenen Sektionen des Bildes oder an verschiedenen Frames arbeiten. Dieser Abschnitt beinhaltet eine vollständige Erklärung der Eigenschaften von Blender, bezogen auf den Herstellungsprozess von Bildern und Animationen.

Nach dem Sie Materialien, Texturen, Belichtung und die Kamera eingerichtet haben, können Sie mit dem Rendern beginnen. Es ist unwahrscheinlich dass Sie es im ersten Anlauf alles richtig machen, also bereiten Sie sich darauf vor mehrere Test Renderings durchzuführen. Diese Abteilung beschreibt die Optionen und Einstellungen für den Renderprozess, der in Ihrer gewünschten Bildqualität resultiert.

Blender hat eine interne Render Engine welche genutzt wird. Diese ist schnell, und kann wenn sie genau eingestellt ist gute Ergebnisse liefern. Es existieren mehrere andere externe Renderer welche geladen werden können, die erweiterte Rendering Tools anbietet.

Wir wissen dass rund um den Globus, unsere Nutzer PC's mit unterschiedlicher Leistung haben. Rendering ist *der* Prozess in CG der CPU und Speicherplatz auffrisst als ob es kein Morgen gäbe. Besonders in Unternehmen, Especially in corporate environments, it is easy to fill up terabyte servers by uploading ten hour-long DV tapes and doing some editing. So, there are lots of options try to shoehorn a big job into a small PC by providing you with multiple sets of options that chunk up the work as best we can, while still preserving image integrity.

This page discusses the main options found on the Render panel, and subsequent explain more.

Überblick

The rendering of the current scene is performed by pressing the big Image button in the Render panel, or by pressing F12 (you can define how the rendered image is displayed on-screen in the [Render Output Options](#)). See also the [Render Window](#).

A movie is produced by pressing the big Animation button. The result of a rendering is kept in a buffer and shown in its own window. It can be saved by pressing F3 or via the File->Save Image menu using the output option in the Output panel. Animations are saved according to the format specified, usually as a series of frames in the output directory. See [Output Options](#) and [Animations](#).

Das Bild wird anhand der Dimensionen die im Dimensionen Panel angegeben sind gerendert.

Arbeitsablauf

In general, the process for rendering is:

1. Create all the objects in the scene
2. [Light the scene](#)
3. [Position the Camera](#)
4. Render a test image at 25% or so without oversampling or raytracing etc. so that it is very fast and does not slow you down
5. Set and Adjust the materials/textures and lighting
6. Iterate the above steps until satisfied at some level of quality
7. Render progressively higher-quality full-size images, making small refinements and using more compute time
8. Save your images

Verteilte Render Farm

There are several levels of CPU allocation that you can use to decrease overall render time by applying more brainpower to the task.

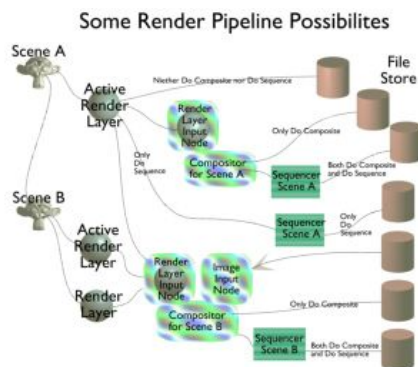
First, if you have multi-core CPU, you can increase the number of threads, and Blender will use that number of CPUs to compute the render.

Second, if you have a local area network with available PC's, you can split the work up by frames. For example, if you want to render a 200 frame animation, and have 5 PC's of roughly equal processing power, you can allocate PC#1 to produce frames 1-40, PC#2 to frames 41-80, and so on. If one PC is slower than the others, simply allocate fewer frames to that PC. To do LAN renders, map the folder containing the .blend file (in which you should have packed your external data, like the textures, ...) as a shareable drive. Start Blender on each PC and open the .blend file. Change the Start and End frame counts on that PC, but do not save the .blend file. Start Rendering. If you use relative paths for your output pathspec, the rendered frames will be placed on the host PC.

Third, you can do WAN rendering, which is where you email or fileshare or Verse-share the .blend file (with packed datas!) across the Internet, and use anyone's PC to perform some of the rendering. They would in turn email you the finished frames as they are done. If you have reliable friends, this is a way for you to work together.

Fourth, you can use a render farm service. These service, like BURP, is run by an organization. You email them your file, and then they distribute it out across their PC's for rendering. BURP is mentioned because it is free, and is a service that uses fellow Blender users PC's with a BOINC-type of background processing. Other services are paid subscription or pay-as-you-go services.

Render Workbench Integration



Blender has three independent rendering workbenches which flow the image processing in a pipeline from one to the other in order:

- Rendering Engine
- [Compositor](#)
- [Sequencer](#)

You can use each one of these independently, or in a linked workflow. For example, you can use the Sequencer by itself to do post processing on a video stream. You can use the Compositor by itself to perform some color adjustment on an image. You can render the scene, via the active Render Layer, and save that image directly, with the scene image computed in accordance with the active render layer, without using the Compositor or Sequencer. These possibilities are shown in the top part of the image to the right.

You can also link scenes and renders in Blender as shown, either directly or through intermediate file storage. Each scene can have multiple render layers, and each Render Layer is mixed inside the Compositor. The active render layer is the render layer that is displayed and checked active. If the displayed render layer is not checked active/enabled, then the next checked render layer in the list is used to compute the image. The image is displayed as the final render if Compositing and Sequencer are NOT enabled.

If Compositing is enabled, the render layers are fed into the Compositor. The noodles manipulate the image and send it to the Composite output, where it can be saved, or, if *Do Sequence* is on, it is sent to the Sequencer.

If Sequencer is enabled, the result from the compositor (if *Do Composite* is enabled) or the active Render layer (if *Do Composite* is not enabled) is fed into the Scene strip in the Sequencer. There, it is manipulated according to the VSE settings, and finally delivered as the image for that scene.

Things get a little more complicated when a .blend file has multiple scenes, for example Scene A and Scene B. In Scene B, if Compositing is enabled, the Render Layer node in Scene B's compositor can pull in a Render Layer from Scene A. Note that this image will not be the post-processed one. If you want to pull in the composited and/or sequenced result from Scene A, you will have to render Scene A out to a file using Scene A's compositor and/or sequencer, and then use the Image input node in Scene B's compositor to pull it in.

The bottom part of the possibilities graphic shows the ultimate blender: post-processed images and a dynamic component render layer from Scene A are mixed with two render layers from Scene B in the compositor, then sequenced and finally saved for your viewing enjoyment.

These examples are only a small part of the possibilities in using Blender. Please read on to learn about all the options, and then exercise your creativity in developing your own unique workflow.

The Render Settings Panel

The Render tab contains all of the options for the internal render engine, or an external one, if selected.

Render

Here you can activate the rendering process, by rendering a [Still Image](#) or an [Animation](#)

You can also select where the image is rendered to. These are described on the [Render Display](#) page.

Layers

The Layers menu contains options for rendering in [Layers](#) and [Passes](#)

Dimensions

This menu has settings for the size of the rendered images (see [Output Options](#)), and options for rendering sequences (see [Animations](#)).

Anti-Aliasing

[Antialiasing](#) is important for producing high quality renders that do not have "jaggies" or stair-stepped pixel artifacts.

Motion Blur

[Motion Blur](#) is an important effect in rendering moving images. It prevents the animation from appearing unrealistic and stuttery, as in stop-motion, where each frame is a perfect still image.

Shading

These are options for controlling what shading effects are calculated in the render. Deselecting them disables them.

- [Textures](#)
- [Shadows](#)
- [Subsurface Scattering](#)
- [Environment Maps](#)
- [Ray Tracing](#)
- [Color Management](#)

Uses a linear workflow when enabled

- [Alpha](#)

Set how transparent pixels are rendered

Output

Set where images are rendered to and what file type. See [Output Options](#).

Leistung

Control the renderer performs in respect to the computer's memory and processor. See [Performance](#).

Post Processing

Control effects that are applied after the image has been rendered. If you are using the [Compositor](#) or [Sequencer](#), you can tell Blender to process those effects instead of directly rendering the scene.

Fields are used when [Rendering for Video](#)

[Dithering](#) is method of blurring pixels.

You can also enable [Edge Rendering](#) to create sketchy or toon-like effects.

Stempel

[Stamping](#) inserts text over the rendered images.

Backen

[Render Baking](#) is a process that creates texture files that hold desired rendered effects, like lighting, shadows, or color information. This is useful for working with real-time graphics that benefit from not having to calculate shading when not necessary.

Anti-Aliasing

Ein vom Rechner generiertes Bild besteht aus Pixeln, diese Pixel können natürlich nur aus einer einzigen Farbe bestehen. Während des Render-Vorgangs muss die Rendering Engine deswegen jedem Pixel auf Basis vom momentan zu sehenden Objekt, jedem Pixel eine einzelne Farbe zuweisen. Das führt oft zu schlechten Ergebnissen, besonders bei scharfen Kanten, oder wo dünne Linien vorhanden sind, und es ist bei schrägen Linien besonders offensichtlich.

Um dieses Problem zu lösen, welches *Aliasing* genannt ist, ist es möglich auf eine Anti-Aliasing Technik zurückzugreifen. Grundsätzlich, wird jedes Pixel 'überabgetastet', durch das Rendern als ob es 5 oder mehr sind, und die Zuweisung einer 'durchschnittlichen' Farbe zu dem gerenderten Pixel.

Die Knöpfe zur Kontrolle von Anti-Aliasing, oder Überabtastung (OSA), sind unterhalb der Render Knöpfe im Render Panel (*Render Panel*).

Optionen

Anti-Aliasing Kontrollkästchen
Aktiviert Überabtastung

5 / 8 / 11 / 16

Die Anzahl an Abtastungen. Die Werte 5, 8, 11, 16 sind vorher-gesetzte Nummern in speziellen Abtastungsmustern;; ein höherer Wert erzeugt bessere Kanten, aber verlangsamt den Render-Vorgang.

Standardmäßig, we use in Blender a fixed "Distributed Jitter" table. The samples within a pixel are distributed and jittered in a way that guarantees two characteristics:

1. Each sample has equal distances to its neighbor samples
2. The samples cover all sub-pixel positions equally, both horizontally and vertically

The images below show Blender sample patterns for 5, 8, 11 and 16 samples. To show that the distribution is equalized over multiple pixels, the neighbor pixel patterns were drawn as well. Note that each pixel has an identical pattern.

5 samples

8 samples

11 samples

16 samples

Full Sample

For every anti-aliasing sample, save the entire Render Layer results. This solves anti-aliasing issues with compositing.

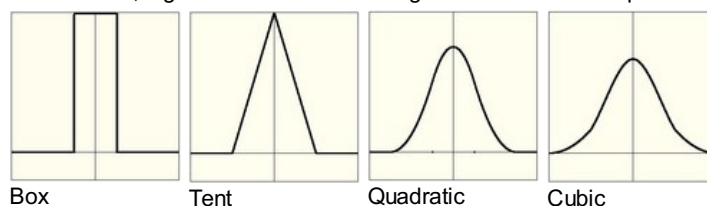
Filtering

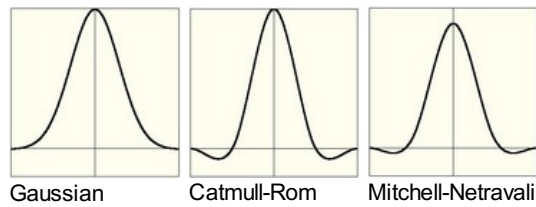
When the samples have been rendered, we've got color and alpha information available per sample. It then is important to define how much each sample contributes to a pixel.

The simplest method is to average all samples and make that the pixel color. This is called using a "Box Filter". The disadvantage of this method is that it doesn't take into account that some samples are very close to the edge of a pixel, and therefore could influence the color of the neighbor pixel(s) as well.

Filter menu: Set The filter type to use to 'average' the samples: |Box|The original filter used in Blender, relatively low quality. For the Box Filter, you can see that only the samples within the pixel itself are added to the pixel's color. For the other filters, the formula ensures that a certain amount of the sample color gets distributed over the other pixels as well.

Box	A low quality box shaped curve (see above)
Tent	A simplistic filter that gives sharp results
Quadratic	A Quadratic curve
Cubic	A Cubic curve
Gauss	Gaussian distribution, the most blurry
Catmull-Rom	Catmull-Rom filter, gives the most sharpening
Mitchell-Netraval	Mitchell-Netraval, a good all around filter that gives reasonable sharpness

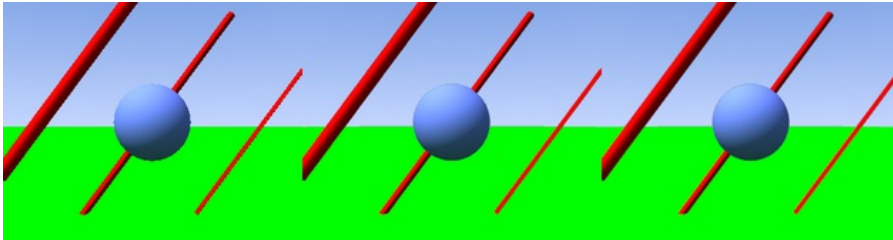




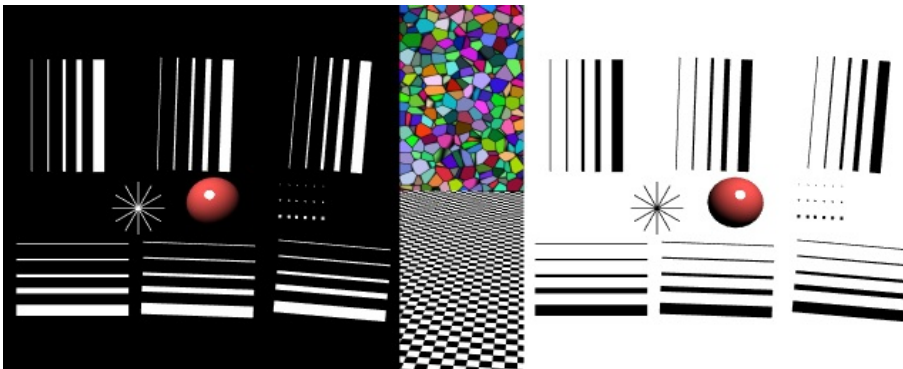
Filter Size

Making the filter size value smaller will squeeze the samples more into the center, and blur the image more. A larger filter size make the result sharper. Notice that the last two filters also have a negative part, this will give an extra sharpening result.

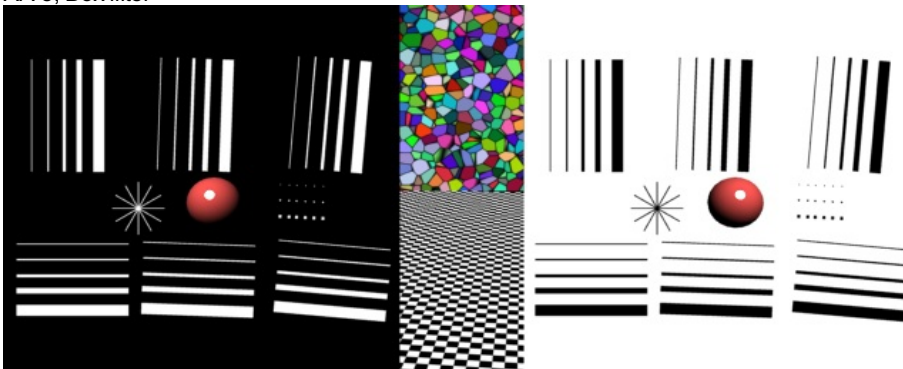
Examples



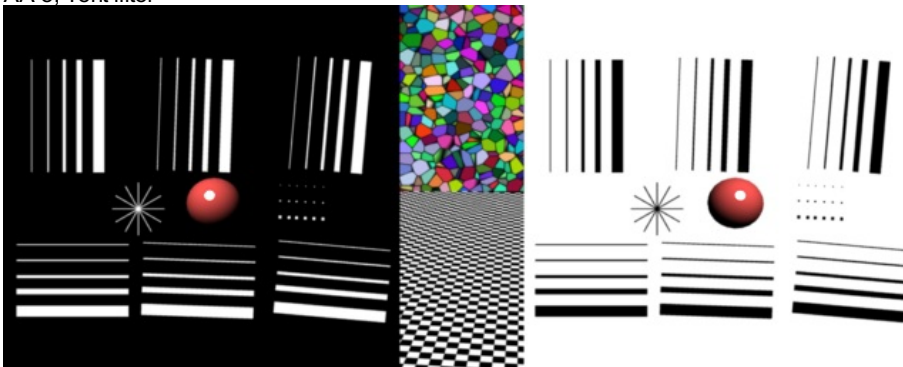
Rendering without AA (left) with AA=5 (center) and AA=8 (right).



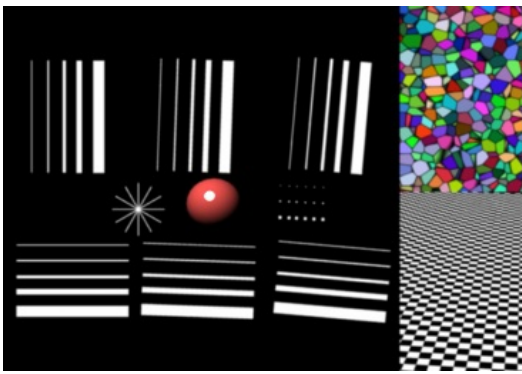
AA 8, Box filter



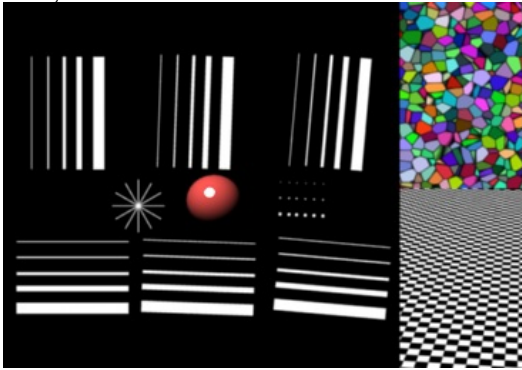
AA 8, Tent filter



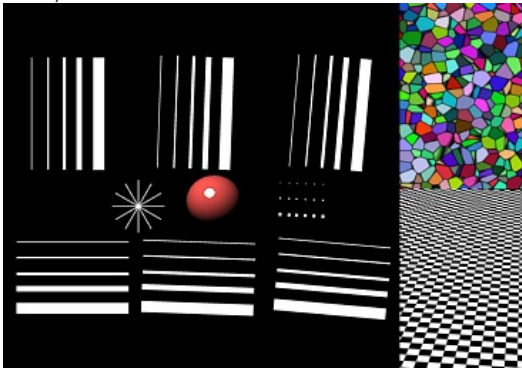
AA 8, Quadratic filter



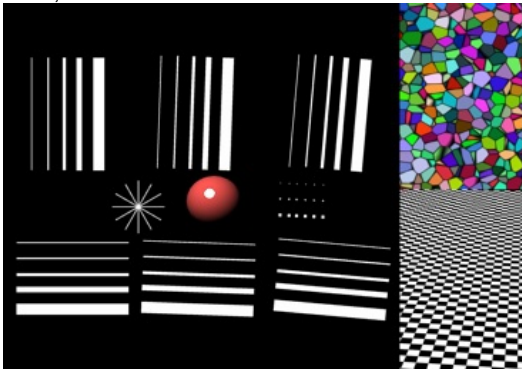
AA 8, Cubic filter



AA 8, Gaussian filter



AA 8, Catmull-Rom filter



AA 8, Mitchell-Netravali filter

Bewegungsunschärfe

Animationen mit Blender werden normalerweise als Reihe von perfekt stillstehenden Bildern gerendert. Das ist unrealistisch, da schnelle Objekte normalerweise, sowohl in einem Einzelbild einer Videoaufnahme als auch in einem Foto, durch ihre Bewegung verschwommen erscheinen. Um einen solchen Bewegungsunschärfeneffekt zu verwenden kann Blender den eigentlichen Frame und noch zusätzliche Zwischenframes bis zum nächsten eigentlichen Frame rendern und diese vermischen um schnell bewegte Details verschwimmen zu lassen.

Das Menschliche Auge

Ein Gehirn verarbeitet etwa 15 Bilder von jedem Auge parallel in einer Sekunde. Diese Bilder werden im Gehirn miteinander verglichen und Bewegung wird festgestellt. Wenn sich etwas sehr schnell bewegt, sieht man den Gegenstand verschwommen (weil die Augen dem Gegenstand mit Verzögerung folgen und weil die Zeitspanne in der der Gegenstand quasi an der selben Stelle ist zu kurz ist um für das Gehirn als Einzelbild erkennbar zu sein). Das einzig Wichtige daran ist, dass ich eine Bewegungsunschärfe sehe.

Im Film

To keep us from seeing jumpy motion pictures, we simply doubled the frame rate to 30 frames per second (fps) (24 fps EU). So, the shutter is basically open for a 30th of a second and the film is exposed to the world for that length of time. As things moved in the real world during that time, the film exposure caused the image of the moving thing to be physically blurred or smeared on that frame. When developed and shown, we physically see an image that is blurred. The POINT IS, I see a blurred image.

In der Computergrafik

In CG, when a frame is rendered, the computer knows exactly where everything should be, and renders it as such. From frame to frame, an object is location A in frame 1, and location B in frame 2. When we show you these two frames at speed (30 fps), the image appears jumpy to us, because, somewhere between the eyeballs and the film, there isn't that same blurring as the real world and film, and we can tell.

Bewegungsunschärfe in Blender

Wie kann man ein computergeneriertes Bild unscharf machen? In Blender gibt es zwei Möglichkeiten, Bewegungsunschärfe zu erzeugen:

Abtastbewegungsunschärfe

Dieses Verfahren ist langsam, aber es liefert bessere Ergebnisse. Es kann in der motion blur section im render options panel aktiviert werden.

Motion Samples

Die Anzahl der Samples für ein Bild

Shutter

Die Zeit, die zwischen Öffnen und schließen des Verschlusses vergeht

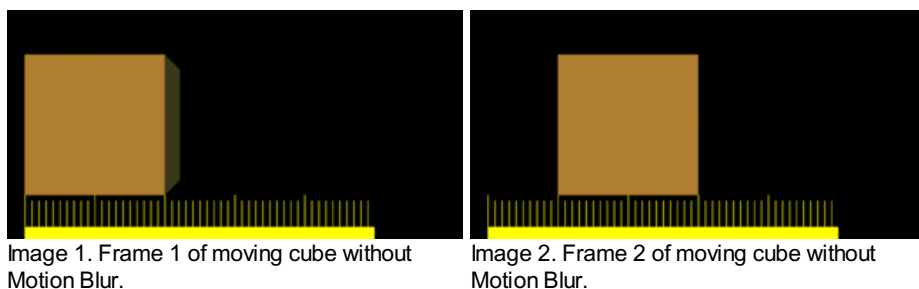
Vektoren Bewegungsunschärfe

[Vector Blur](#) ist schneller, kann aber zu unerwünschten Nebeneffekten führen - die allerdings verhindert werden können. Vector blur wird im Compositing durchgeführt, indem die Szene ohne Unschärfe und mit einen Pass, der die Bewegungsinformationen für jeden Pixel enthält, gerendert wird. Diese Information ist ein Vektor, der eine 3D Richtung und einen Betrag enthält. Der Compositor nutzt diese Information, um Unschärfe auf jeden Pixel in die angegebene Richtung anzuwenden.

Beispiele

To better grasp the concept let's assume that we have a cube, uniformly moving 1 Blender unit to the right at each frame. This is indeed fast, especially since the cube itself has a side of only 2 Blender units.

Image 1. shows a render of frame 1 without Motion Blur, *Image 2.* shows a render of frame 2. The scale beneath the cube helps in appreciating the movement of 1 Blender unit.



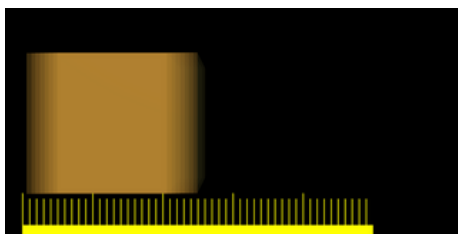


Image 3. Frame 1 of moving cube with Motion Blur, 8 samples, Shutter=0.5.

Image 3. on the other hand shows the rendering of frame 1 when Motion Blur is set and 8 'intermediate' frames are computed. Shutter is set to 0.5; this means that the 8 'intermediate' frames are computed on a 0.5 frame period starting from frame 1. This is very evident since the whole 'blurriness' of the cube occurs on half a unit before and half a unit after the main cube body.

Image 4. and *Image 5.* show the effect of increasing Bf values. A value greater than 1 implies a very 'slow' camera shutter.

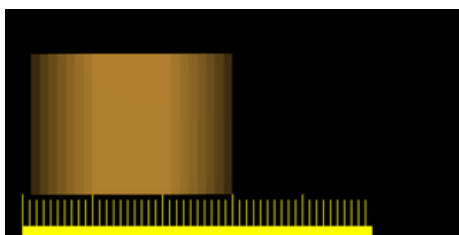


Image 4. Frame 1 of moving cube with Motion Blur, 8 samples, Shutter=1.0.

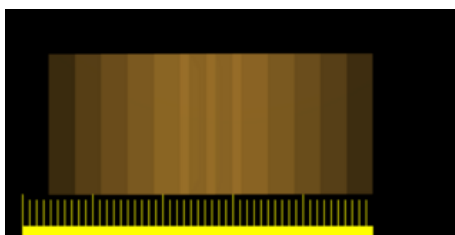


Image 5. Frame 1 of moving cube with Motion Blur, 8 samples, Shutter=3.0.

Better results than those shown can be obtained by setting 11 or 16 samples rather than 8, but, of course, since as many *separate* renders as samples are needed a Motion Blur render takes that many times more than a non-Motion Blur one.

Hinweise

If Motion Blur is active, even if nothing is moving on the scene, Blender actually 'jitters' the camera a little between an 'intermediate' frame and the next. This implies that, even if Anti-Aliasing is off, the resulting images have nice Anti-Aliasing. An MBLUR obtained Anti-Aliasing is comparable to an Anti-Aliasing of the same level, but generally slower.

This is interesting since, for very complex scenes where a level 16 Anti-Aliasing does not give satisfactory results, better results can be obtained using *both* Anti-Aliasing and MBlur. This way you have as many samples per frame as you have 'intermediate' frames, effectively giving oversampling at levels 25,64,121,256 if 5,8,11,16 samples are chosen, respectively.

Page status ([reviewing guidelines](#))

Void page

Proposed fixes: none

Subpages

- [Cycles](#)
 - [Devices](#)
 - [Erste Schritte](#)
- [Freestyle](#)

Cycles

Cycles ist eine neue Renderengine welche ab der Blenderversion 2.61 in Blender fest integriert ist. Cycles wird ständig weiterentwickelt wobei beabsichtigt ist, eine benutzerfreundliche und interaktiv zu bedienende Renderengine zur Verfügung zu stellen welche trotzdem über viele Features verfügt.

[200px](#)

Erste Schritte

Cycles ist als Addon integriert welches standardmäßig aktiviert ist. Um Cycles zu verwenden muß es als aktive Renderengine eingestellt sein.

[Mehr ...](#)

Page status ([reviewing guidelines](#))

Void page

Proposed fixes: [X](#)

Cycles Renderengine

Cycles ist eine Renderengine die es seit Blender 2.61 gibt. Sie wird immer noch entwickelt und soll eine Renderengine mit besonderem Fokus auf Interaktivität und leichter Benutzung werden, während sie dennoch viele Produktionsfunktionen haben soll. Eine [Entwicklungs-Dokumentation](#) ist verfügbar.

Erste Schritte

Cycles ist als ein Addon eingebunden, welches standardmäßig aktiviert ist. Um Cycles zu verwenden, muss Cycles als aktive Renderengine in der Kopfleiste eingestellt werden. Nachdem das getan ist, kann das interaktive Rendern in der 3D-Ansicht genutzt werden, indem der Zeichenmodus auf Rendered gesetzt wird. Der Render wird immer aktualisiert, wenn Materialien oder Objekte verändert wurden.

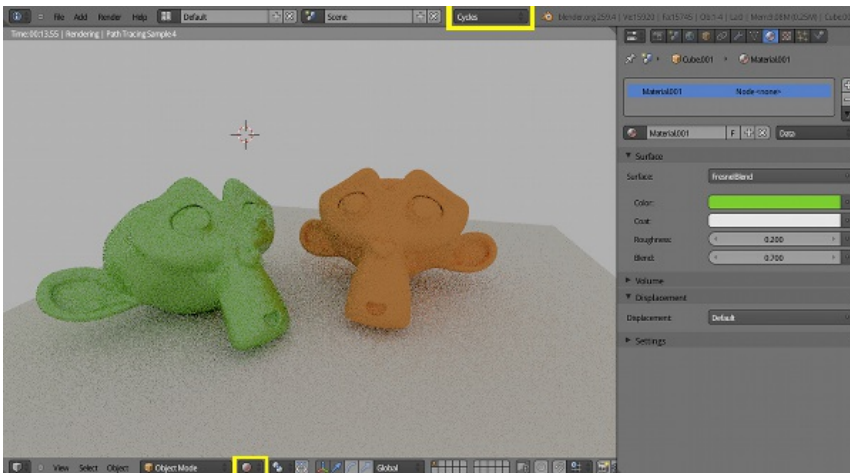
Tutorials

Referenz

- [Kamera](#)
- [Materialien](#)
 - [Oberfläche](#)
 - [Volumen](#)
 - [Verformung](#)
- [Welt](#)
- [Lampen](#)
- [Nodes](#)
 - [Shader](#)
 - [Texturen](#)
 - [Mehr](#)
- [Licht Pfade](#)
- [Integrator](#)
- [Rauschreduzierung](#)
- [Render Durchläufe](#)
- [Texturbearbeitung](#)
- [GPU Rendering](#)

Vor dem Start

Cycles ist als gebündeltes Addon standardmäßig aktiviert. Um Cycles zu nutzen, muss es als die aktive Renderengine in der Kopfzeile vom Blender Hauptfenster aktiviert sein. Wenn dass erledigt ist, kann die interaktive Bearbeitung durch das Setzen eines 3D Ansichts Editors auf den Zeichenmodus begonnen werden. Der Render aktualisiert sich sobald Material und Objekt Veränderungen abgeschlossen sind.

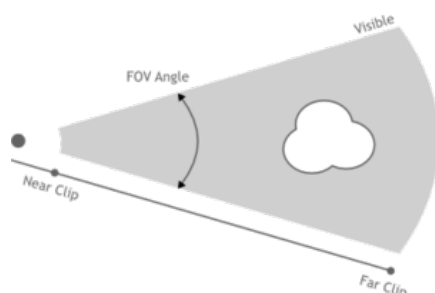


Um zu sehen ob und wie Sie ihre GPU für das Rendering nutzen können, siehe [GPU Rendering](#).

Kamera

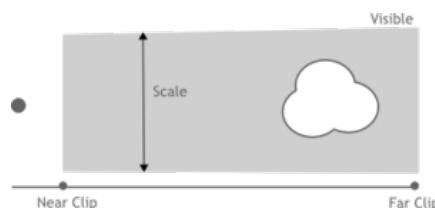
Perspektivisch

Objektivgröße und Winkel
Veränderung des Sichtwinkels.



Orthografisch

Skalieren
Kontrolliert die Größe der projizierten Objekte im Bild.



Panorama

Diese Option funktioniert ein wenig anders als die standard Blenderkamera und kann nicht im OpenGL-Render angezeigt werden. Es rendert eine Panoramansicht der Szene von der Kameraposition und verwendet eine äquirektanguläre Projektion, wobei immer die vollen 360° der X-Achse und 180° über die Y-Achse berechnet werden.

Diese Projektion ist mit der Umgebungstextur kompatibel, welche für den World-Shader verwendet wird. Deshalb kann sie auch als Umgebungstextur genutzt werden. Um die vorgegebene Abbildung zu erreichen, sollte die Objektrotation der Kamera um (90, 0, -90) verändert werden oder entlang der positiven X-Achse zeigen. Dies entspricht dem Schauen auf das Zentrum des Bildes bei einer normalen Umgebungstextur.

Tiefenunschärfe

Aperture Type

Methode, mit der die Größe der Blendenöffnung angegeben wird. Mit Radius kann der Radius der Öffnung angegeben werden, während F/Stop die Größe relativ zur Kamerabrennweite angibt, ein übliches Maß in der Fotografie. Es gilt die Beziehung:

$$\text{Blendenradius} = \text{Brennweite} / (2 \text{ f-stop})$$

Aperture Size

auch Linsenradius genannt. Wenn dieser Wert 0 ist, erscheinen alle Objekte im Fokus, während größere Werte die Objekte entfernt von der Brennweite außerhalb des Fokus erscheinen lässt.

Aperture F/Stop

auch F-number oder relative Blende genannt. Kleinere Werte geben mehr Tiefenunschärfe, größere Werte geben ein schärferes Bild.

Aperture Blades

Wenn dieser Wert 3 oder höher ist, wird eine vieleckig geformte Blende anstelle eines Kreises verwendet, was dann den Eindruck der Lichter außerhalb des Fokus beeinflusst.

Aperture Rotation

Rotation der Aperture Blades.

Focal Distance

Distanz, wo Objekte in perfektem Fokus sind. Alternativ kann ein Objekt festgelegt werden, wessen Distanz von der Kamera genommen wird.

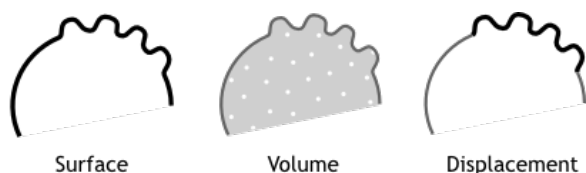
Clipping

Clip Start and End

Dies ist der Intervall, in dem Objekte sichtbar sind. Alle Objekte außerhalb dieses Bereichs beeinflussen das Bild indirekt, da Lichtstrahlen nicht abgeschnitten werden. Für das OpenGL-Rendern sind die Clipping-Werte wichtig, um ausreichend Rasterungspräzision sicherzustellen. Raytracing leidet nicht sehr unter diesem Aspekt, weshalb extremere Werte sicher eingestellt werden können.

Materialien

Materialien definieren die Erscheinung des Meshs, von Kurven und anderen Objekten. Sie bestehen aus drei Shadern, beschreiben das Aussehen der Oberfläche, des Körperinhaltes (Volume) und der Verformung (Displacement) der Oberfläche des Meshs.



Surface Shader

Der Surface Shader beschreibt das Zusammenwirken des Lichtes mit der Oberfläche. Eine oder mehrere BSDF's legen fest ob auftreffendes Licht reflektiert wird, in das Mesh gebrochen oder absorbiert wird.

Emission bestimmt, wie Licht von der Oberfläche emittiert wird. Dies ermöglicht, dass jede Fläche als Lampe fungiert.

Volume Shader

Der Volume Shader befindet sich derzeit unter unabhängiger Entwicklung.

Wenn der Surface Shader Licht nicht reflektiert oder absorbiert, geht es in den Raum des Meshs. Wenn kein Volume Shader bestimmt wurde, wird es direkt durch das Mesh zur anderen Seite durchgehen.

Wenn es festgelegt wurde, beschreibt ein Volume Shader die Interaktion des Licht, wenn es durch den Rauminhalt des Meshes geht. Licht kann gestreut, absorbiert oder an jedem Punkt im Raum emittiert werden.

Ein Material kann sowohl einen Surface- als auch einen Volume Shader haben, oder nur einen von beiden. Die Verwendung beider kann für Materialien wie Glas, Wasser oder Eis nützlich sein, wo ein Teil des Lichtes absorbiert werden soll, während es durchgeht, kombiniert mit einem Glass- oder Glossy Shader auf der Oberfläche.

Displacement

Die Form der Oberfläche und der Raum innen können durch Displacement (Verformung) verändert sein. Über diesen Weg können mit der Verwendung von Texturen Meshoberflächen detaillierter erscheinen. Abhängig von den Einstellungen kann das Displacement virtuell (verändert nur die Oberflächen-Normalen um den Eindruck von Verformung zu erreichen; bekannt als Bumpmapping) oder als Kombination aus echter und virtueller Verformung verwendet werden.

Energieeinsparung

Das Materialsystem ist mit dem Hintergrund physikalisch basierten Renderings entwickelt worden, wobei die Erscheinung eines Materials und der verwendete Renderalgorithmus klar getrennt sind. Das macht es leichter realistische Ergebnisse und eine ausgewogene Beleuchtung zu erhalten, wobei ein paar Dinge zu beachten sind.

Damit Materialien gut mit der Globalen Illumination arbeiten, sollten sie, physikalisch gesprochen, energiesparend sein. Das heißt, sie können nicht mehr Licht reflektieren, als auftrifft. Diese Eigenschaft ist nicht strikt erzwungen, aber wenn Farbwerte im Bereich zwischen 0.0 und 1.0 liegen und BSDF's nur durch Mix Shader gemischt werden, wird dies automatisch erfüllt.

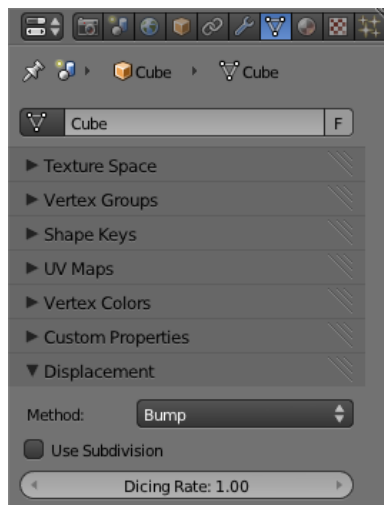
Dennoch ist es möglich, dies mit höheren Farbwerten als 1.0 oder dem Add Shader zu brechen. Deshalb sollte man dabei vorsichtig sein, um das Verhalten der Materialien unter verschiedenen Beleuchtungssituationen durchschaubar zu halten. Es kann in einer Reflektion enden, die Licht bei jedem Durchlauf addiert, was einen BSDF in eine Art Emitter verändert.

Verformung

Noch nicht fertig implementiert, als [experimentelles Feature](#) markiert.

Die Form der Oberfläche und der Raum innen können durch Verformung verändert sein. Über diesen Weg können mit der Verwendung von Texturen Meshoberflächen detaillierter erscheinen.

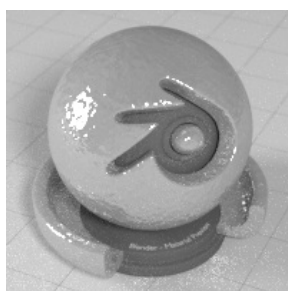
Typ



Abhängig von den Einstellungen kann die Verformung virtuell (verändert nur die Oberflächen-Normalen um den Eindruck von Verformung zu erreichen; bekannt als Bumpmapping) oder als Kombination aus echter und virtueller Verformung verwendet werden. Die Optionen zum Verformen sind:

- **Echte Verformung:** Mesh Vertices werden vor dem Rendern angezeigt und das aktuelle Mesh verändert. Dies gibt die besten Resultate, wenn das Mesh fein unterteilt ist. Die Folge ist, dass dieses Verfahren am speicherintensivsten ist.
- **Bump Mapping:** Bei Ausführung wird beim Surface Shader eine veränderte Oberflächennormale anstatt der echten Normale verwendet. Dies ist eine schnelle Alternative zu echter Verformung, ist aber nur eine Näherung. Oberflächenformen werden nicht fein und es gibt keine Eigenschaften der Verformung.
- **Verformung + Bump:** Beide Methoden sind kombinierbar um Verformung auf einem gröberen Mesh durchzuführen und Bump Mapping für feine Details zu nutzen.

Unterteilen



Bump Mapped Verformung

Noch nicht fertig implementiert, als [experimentelles Feature](#) markiert.

Für detaillierte Verformung muss das Mesh in kleinere Polygone unterteilt werden. Dies kann durch das Hinzufügen eines Unterteiler-Oberflächen-Modifikators erreicht werden, aber es ist auch möglich, die Renderengine das Mesh unterteilen zu lassen.

Surface

Der Surface Shader beschreibt das Zusammenwirken des Lichtes mit der Oberfläche. Eine oder mehrere BSDF's legen fest ob auftreffendes Licht reflektiert wird, in das Mesh gebrochen oder absorbiert wird.

Emission bestimmt, wie Licht von der Oberfläche emittiert wird. Dies ermöglicht, dass jede Fläche als Lampe fungiert.

Begriffe

- **BSDF** steht für Bidirectional Scattering Distribution Function. Es definiert, wie Licht auf einer Oberfläche absorbiert und gebrochen wird.
- **Reflection** BSDF's reflektiert einen einkommenden Strahl auf der gleichen Seite des Meshs.
- **Transmission** BSDF's sendet einen einkommenden Strahl durch das Mesh, welches es auf der anderen Seite verlässt.
- **Refraction** BSDF's ist eine Variante des **Transmission**, welches eine einfallenden Strahl leitet und dessen Richtung verändert, wenn er die andere Seite der Oberfläche verlässt.

BSDF Parameter

Ein bedeutender Unterschied zu nicht physikalisch basierten Renderern ist, dass direkte Lichtreflektionen von Lampen und indirekte Lichtreflektionen von anderen Oberflächen nicht entkoppelt, sondern eher zusammen durch die Verwendung eines BSDF behandelt werden. Dies beeinträchtigt die Möglichkeiten ein wenig, aber wir denken, dass es insgesamt hilfreich bei der Erstellung einheitlich aussehender Render mit weniger Parametern zum Verbessern ist.

Für glatte BSDF's kontrolliert der **roughness**-Parameter die Härte der Reflektionen von 0.0 für perfekte Schärfe bis 1.0 für sehr weiche. Verglichen zu **Härte**- oder **Exponent**-Parametern hat es den Vorteil im Bereich von 0.0 bis 1.0 zu sein, resultierend linearere Kontrolle zu geben und leichter texturierbar zu sein. Diese Beziehung ist ungefähr: $Rauheit = 1 - 1/Härte$

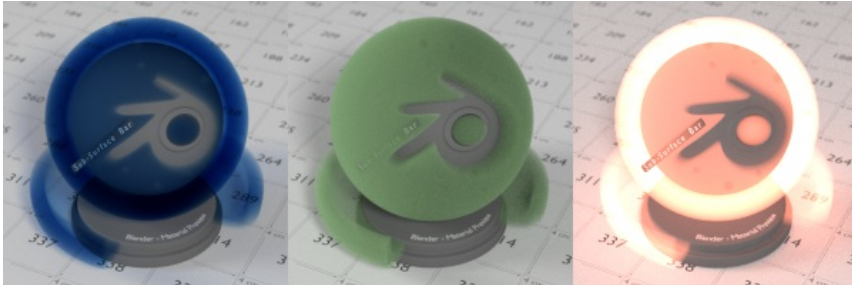
Volume

Der Volume Shader befindet sich derzeit unter unabhängiger Entwicklung.

Wenn der Surface Shader Licht nicht reflektiert oder absorbiert, geht es in den Raum des Meshs. Wenn kein Volume Shader bestimmt wurde, wird es direkt durch das Mesh zur anderen Seite durchgehen.

Wenn es festgelegt wurde, beschreibt ein Volume Shader die Interaktion des Licht, wenn es durch den Rauminhalt des Meshes geht. Licht kann gestreut, absorbiert oder an jedem Punkt im Raum emittiert werden.

Ein Material kann sowohl einen Surface- als auch einen Volume Shader haben, oder nur einen von beiden. Die Verwendung beider kann für Materialien wie Glas, Wasser oder Eis nützlich sein, wo ein Teil des Lichtes absorbiert werden soll, während es durchgeht, kombiniert mit einem Glass- oder Glossy Shader auf der Oberfläche.



Der Volume Shader

World

Die Weltumgebung kann Licht emittieren, als einzelne Farbe, als physikalisch korrekter Himmel oder als beliebige Textur.

Surface Shader

Der Surface Shader bestimmt die Lichtemission von der Umgebung in die Szene. Der Welthintergrund wird gerendert, als wäre er sehr weit von der Szene entfernt. Deshalb gibt es keine zweiseitige Interaktion zwischen Objekten in der Szene und der Umgebung, nur Licht kommt hinein. Der einzige akzeptierte Shader ist die Background-Node mit einer Farbe als Eingabe und der Stärke-Faktor für die Intensität des Lichtes.

Volume Shader

Unter unabhängiger Entwicklung

Ambient Occlusion

Ambient occlusion ist eine Beleuchtungsmethode, abhängig, wie stark ein Punkt auf einer Fläche von anderen Flächen beeinflusst wird. Dies ist ein nicht physikalisch korrekter Trick, ist aber hilfreich um Formen von Oberflächen zu betonen, oder um auf einem einfachen Weg einen Effekt zu erhalten, der indirekter Beleuchtung ähnelt.

- Factor: Die Stärke der AO, der Wert 1.0 ist wie ein weißer World Shader.
- Distance: Distanz eines Punktes, in der ein Strahl verfolgt wird. Shorter distance emphasizes nearby features Kürzere Distanzen betonen kleinere Stellen, weitere Distanzen führen zur Beeinflussung weiter entfernter Objekte.

Die Beleuchtung durch AO wird nur auf diffus reflektierende BSDF's angewand, glänzende oder durchscheinende BSDF's werden nicht berücksichtigt. Transparenz von Oberflächen wird berücksichtigt. Zum Beispiel wird eine halbtransparente Oberfläche nur halb okkludiert.

Tricks

Manchmal kann es nützlich einen anderen sichtbaren Hintergrund, als den für die indirekte Beleuchtung der Objekte. Eine einfache Lösung ist das Hinzufügen einer Mix-Node mit dem Blend-Factor "Is Camera Ray". Der erste Input ist dann die sichtbare Farbe und der zweite Input ist die indirekte sichtbare Farbe.

Background Shader sind zur Zeit nicht "importance sampled", das heißt, dass sie verwenden nicht mehr Samples in Regionen des Hintergrundes, die heller sind. Dies wiederum kann zu einer langsameren Annäherung mit Texturen führen, die helle Lichter haben. Solche Texturen weich zu zeichnen kann das Bildrauschen reduzieren und schneller rauschfrei sein.

Lampen

Neben der Beleuchtung durch die Umgebung und Objekte mit einem Emission-Shader sind Lampen ein weiterer Weg, um Licht zur Szene hinzuzufügen. Der Unterschied ist, dass sie im gerenderten Bild nicht direkt sichtbar sind und leichter als Objekte eigenen Typs verwaltet werden können.

Type

Zur Zeit werden **Point**, **Area** und **Sun** Lampen unterstützt. Spots und Hemilampen werden nicht unterstützt und als Punkt beziehungsweise als Sun-Lampen gerendert. Eventuell werden sie in der Zukunft funktionieren. Deshalb ist es das Beste, diese nicht zu aktivieren, um Kompatibilität zu bewahren.

Size

Größe der Lampe in Blendereinheiten. Eine Erhöhung führt zu weicheren Schatten.

Cast Shadow

Durch das Deaktivieren dieser Option wird Licht von Lampen nicht durch Objekte dazwischen blockiert. Dies kann das Rendern beschleunigen, da Strahlen zu Lichtquellen nicht verfolgt werden müssen.

Point Lamp

Point-Lampen senden gleiches Licht in alle Richtungen. Durch das Verändern des Size-Wertes größer als null werden die Lampen kugelförmig, was zu weicheren Schatten führt.

Area Lamp

Area-Lampen emittieren Licht von einer quadratischen oder rechteckigen Fläche mit Lambertverteilung.

Sun Lamp

Sun-Lampen emittieren Licht in eine gegebene Richtung. Ihre Position wird nicht beachtet, das heißt, sie sind immer unendlich weit außerhalb der Szene und haben keine Lichtabnahme bei unterschiedlichen Entfernungen.

Da sie nicht in der Szene liegen, verwendet ihrer Stärke andere Einheiten und sollte normalerweise niedrigere Werte als andere Lichter haben.

Nodes

Materialien, Lichter und die Umgebung werden alle durch ein Netzwerk aus Shading-Nodes definiert. Diese Nodes geben Werte, Vektoren, Farben und Shader aus.

Shaders

Ein wichtiges Konzept für den Bau von Node-Gerüsten ist das Verstehen der **Shader-Sockets**. Der Ausgang aller Surface- und Volumeshader ist ein Shader, der sowohl Lichtinteraktionen auf Flächen oder im Raum als auch die Farbe der Flächen beschreibt.

Es gibt wenige, als Nodes verfügbare Shader-Typen:

- **BSDF** Shader beschreibt die Lichtreflektion, -brechung und -absorbierung auf einer Objektoberfläche.
- **Emission** Shader beschreibt die Lichtemission auf einer Objektoberfläche oder im Raum.
- **Volume** Shader beschreibt die Lichtstreuung im Raum.
- **Background** Shader beschreibt die Lichtemission durch die Umgebung.

Jede Shader-Node hat eine Farbeingabe und gibt einen Shader aus. Diese können gemischt oder addiert werden durch einen Mix- oder einen Addshader. Keine andere Operation ist erlaubt. Der resultierende Ausgang kann durch die Renderengine genutzt werden um alle Lichtinteraktionen zu berechnen, für direkte Beleuchtung und globale Illumination.

Texturen

Jede Textur in Cycles entspricht einer Node mit Texturkoordinaten und anderen Parametern als Eingang und einer Farbe als Ausgang. Es werden keine Texturdatenblöcke benötigt. Stattdessen können Node-Gruppen für die Wiederverwendung von Textur-Setups genutzt werden.

Für das UV-Mapping und Texturepainting im Viewport muss die Imagetexture-Node genutzt werden. Wenn solch eine Node aktiviert wird, wird es im Texture-Draw-Mode angezeigt. Dann kann im Texture-Paint-Mode gezeichnet werden.

Normalerweise werden für die Texturkoordinaten die generierten Koordinaten genutzt, mit Ausnahme der Bildtexturen, die standardmäßig UV-Koordinaten verwenden. Jede Node enthält einige Optionen um das Textur-Mapping und die Farben zu verändern. Diese können in den Textureinstellungen verändert werden.

Mehr

Nodes für geometrische Informationen, Texturkoordinaten, Shader mit Schichten und nicht-physikalisch basierte Tricks.

Open Shading Language

Eigene Nodes können mit Hilfe der Open Shading Language geschrieben werden.

Shader Nodes

BSDF

Diffuse

Lambert und Oren-Nayar diffuse Reflektion.

Color input

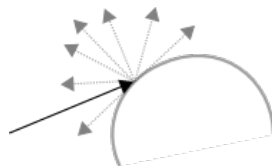
Farbe der Oberfläche, oder physikalisch gesprochen, die Wahrscheinlichkeit, dass Licht für jede Wellenlänge reflektiert oder weitergeleitet wird.

Roughness

Rauhheit der Oberfläche von 0.0 führt zu einer Lambert-Reflektion, höhere Werte aktivieren den Oren-Nayar BSDF.

BSDF output

Diffuser BSDF Shader.



Diffuses Verhalten



Lambertian Diffus



Oren Nayar Diffus

Translucent

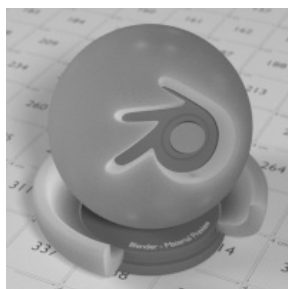
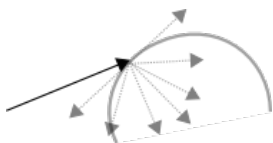
Lambert-Diffuse Übertragung.

Color input

Farbe der Oberfläche, oder physikalisch gesprochen, die Wahrscheinlichkeit, dass Licht jede Wellenlänge übertragen wird.

BSDF output

Lichtdurchlässiger BSDF Shader.



Translucent Shader

Glossy

Glatte Reflektion mit Microfacet-Verteilung, vor allem für Materialien wie Metall oder Spiegel verwendet. Glossy neigt zu vielen Fireflies.

Distribution

Microfacet-Verteilung zu verwenden. Sharp führt zu perfekt spiegelnden Oberflächen wie bei einem Spiegel, während Beckmann und GGX den Roughness-Eingang für verwischte Reflektionen nutzen.

Color input

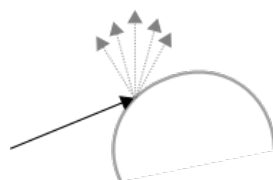
Farbe der Oberfläche, oder physikalisch gesprochen, die Wahrscheinlichkeit, dass Licht bei jeder Wellenlänge reflektiert wird.

Roughness input

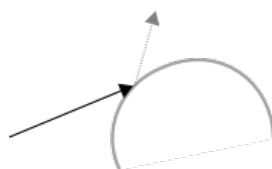
Beeinflusst die Schärfe der Reflektionen, perfekt scharf bei 0,0 und weicher bei höheren Werten.

BSDF output

Glänzender BSDF Shader.



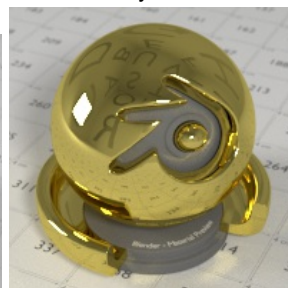
Rauer Glossy-Shader



Glatter Glossy-Shader



Ein raues Glanzmaterial



Ein glattes, glänzendes Material

Transparent

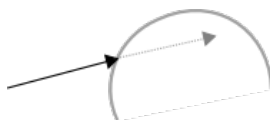
Transparentes BSDF ohne Brechung, wobei das Licht direkt durch die Oberfläche geht, als wenn da keine Geometrie wäre. Dies ist vor allem für Alpha-Maps sinnvoll. Dieser Shader beeinflusst Lichtpfade ein wenig anders als andere BSDF's. Beachte, dass nur reines weiß wirklich komplett transparent ist.

Color input

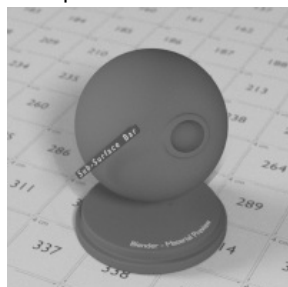
Farbe der Oberfläche, oder physikalisch gesprochen, die Wahrscheinlichkeit für jede Wellenlänge, dass Licht blockiert oder gerade durchgelassen wird.

BSDF output

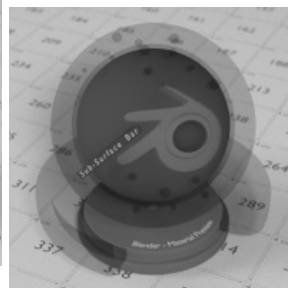
Transparenter BSDF Shader.



Transparentes Verhalten



Transparenter Shader (reines Weiß)



Transparenter Shader (Grau)

Glass

Glasähnlicher Shader, der Lichtbrechung und -reflektion bei bestimmten Winkeln mischt. Ähnlich dem transparenten Shader macht nur reines Weiß den Shader wirklich durchsichtig. der Glas-Shader neigt zu Bildrauschen wegen der Kaustiken.

Distribution

Microfacet-Verteilung wird verwendet. Sharp führt zu perfektem Glas, während Beckmann und GGX den Roughness-Eingang verwenden für raues Glas verwenden können.

Color input

Farbe der Oberfläche, oder physikalisch gesprochen, die Wahrscheinlichkeit, dass Licht für jede Wellenlänge übertragen wird.

Roughness input

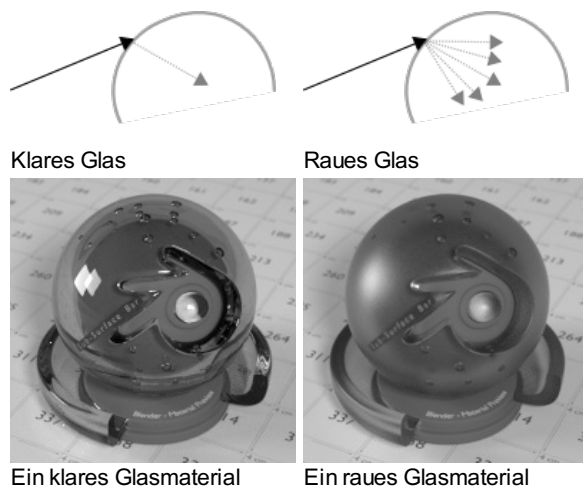
Beeinflusst die Schärfe der Brechnung, perfekte Schärfe bei 0.0 und weichere Schärfe mit höheren Werten.

IOR input

Brechungsindex definiert, wie stark die Strahlenrichtung beeinflusst wird. Bei 1.0 gehen die Strahlen gerade durch wie beim transparenten Shader, während höhere Werte zu stärkerer Brechung führen.

BSDF output

Glas BSDF Shader.



Klares Glas

Raues Glas

Ein klares Glasmaterial

Ein raues Glasmaterial

Velvet

Velvetreflektions-Shader für Materialien wie Kleidung. Zurzeit neigt auch der Velvet-Shader zu Fireflies.

Color input

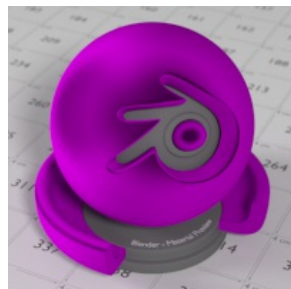
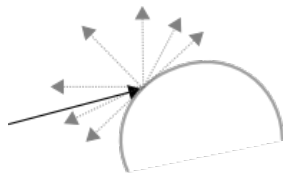
Farbe der Oberfläche, oder physikalisch gesprochen, die Wahrscheinlichkeit, dass Licht bei jeder Wellenlänge reflektiert wird.

Sigma input

keine Beschreibung bisher

BSDF output

Velvet BSDF Shader.



Velvet Shader

Emission

Lambert-Emission für die Verwendung für Materialien und als Lampenausgänge.

Color input

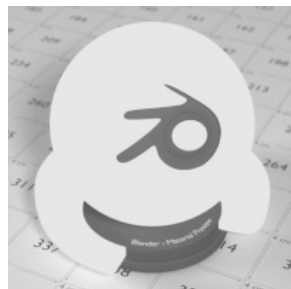
Farbe des emittierten Lichtes.

Strength input

Stärke des emittierten Lichtes. Für Point- und Area-Lampen ist die Einheit Watt. Für Materialien mit dem Wert 1.0 wird das Objekt exakt die gleiche Farbe, wie im Color-Eingang definiert, haben und es 'schattenlos' machen.

Emission output

Emission Shader.



Emittierendes Material mit der Helligkeit 1.0

Background

Diese Node sollte nur für den Word-Surface-Ausgang genutzt werden, da es in anderen Fällen ignoriert wird.

Color input

Farbe des emittierten Lichtes.

Strength input

Stärke des emittierten Lichtes.

Background output

Background Shader.

Holdout

Ein Holdout-Shader ist für Compositing nützlich, da bei Verwendung dieses Shaders ein 'Loch' im Bild mit null Alphantransparenz erzeugt wird.

Holdout output

Holdout shader.



Die schwarze Region hat eine Alphantransparenz von null.

Mix and Add

Mix- oder Add-Shader zusammen. Mischen kann für beschichtete Materialien verwendet werden, wobei der Fac-Eingang zum Beispiel mit einer Blend-Weight-Node verbunden werden kann.

Shader inputs

Shader zum Mischen, sodass einkommende Strahlen mit der im Fac-Eingang angegeben Wahrscheinlichkeit getroffen werden.

Fac input

Blend-Weight zum Mischen zweiter Shader. Bei 0.0 wird nur der erste Shader und bei 1.0 nur der zweite Shader verwendet.

Shader output

Gemischter Shader.



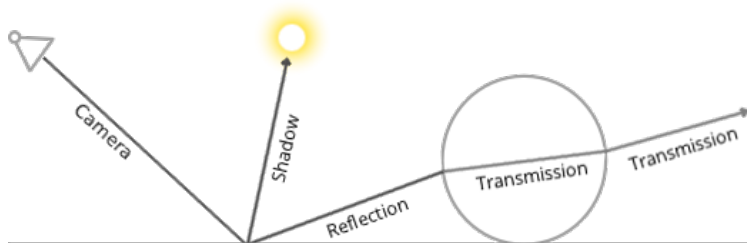
Ein Mix aus einem Glossy-Shader und einem Diffus-Shader liefert ein gutes Keramikmaterial.

Rauschreduzierung

Für schnellere Renderzeiten ist es wichtig, das Rauschen so weit wie möglich zu reduzieren. Hier werden eine Anzahl an Tricks diskutiert, die, während sie die Gesetze der Physik brechen, sehr wichtig für das Rendern von Animationen und Bildern in angemessener Zeit und Qualität sind. Klicken Sie zum Vergrößern auf die Beispielbilder, um die Unterschiede im Rauschen besser zu erkennen.

Path Tracing

Cycles benutzt path tracing mit next event simulation, was beim rendern nicht mit allen Arten von Lichteffekten wie z.B. Kaustiken, gut zurecht kommt. Es hat aber einen Vorteil bei großen und detaillierten Szenen, verglichen mit einigen anderen Renderalgorithmen. Das hat seinen Grund darin, dass keine photon map im Speicher gehalten werden muss und weil die Strahlen im Vergleich zum bidirectional path tracing relativ kohärent im Bildcache gehalten werden können.

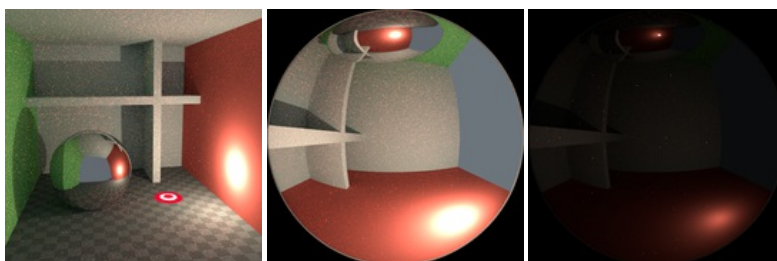


Hier geschieht genau das Gegenteil von dem, was in Wirklichkeit passiert. Anstatt aus den Lampen, kommen die Lichtstrahlen von der Kamera und fallen auf die Szene und die Lichter. Das hat den Vorteil, dass keine Lichtstrahlen berechnet werden müssen, die die Kamera nicht treffen. Aber das heißt ebenso, dass es schwierig ist, Strahlen zu finden, die viel dazu beisteuern können. Die Lichtstrahlen werden entweder zu den Objektoberflächen oder in Richtung der bekannten Lampen gesendet.

Für mehr Informationen, siehe [Light Paths](#) und [Integrator](#) Dokumentation.

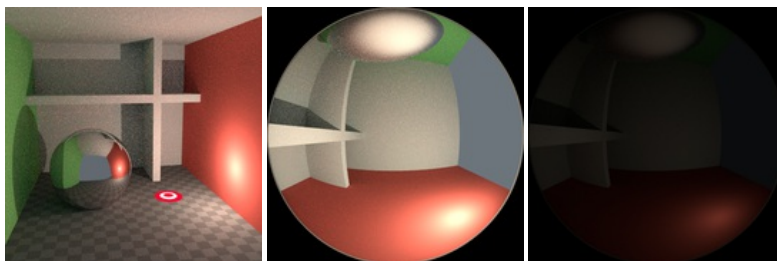
Der Ursprung des Rauschens

Um zu verstehen, wo das Rauschen herkommen kann, betrachten wir folgende Szene. Der diffuse shader sieht die von Oberflächen reflektierten Strahlen und versucht, die durchschnittliche Farbe und Helligkeit aller Pixel zu finden. Die Highlights auf der Kugel und der nahen Wand tragen hier wesentlich zur Beleuchtung bei, da sie um ein vielfaches heller sind als der Rest des Bildes.



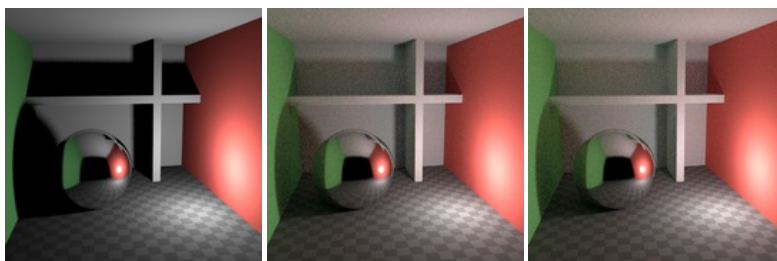
Die Lampe ist als bekannte Lichtquelle nicht allzu schwer zu finden, aber die Highlights, die sie verursacht, machen den Unterschied. Beim path tracing werden die Strahlen gleichmäßig über die Hemishäre verteilt, in der Hoffnung, alle wichtigen hellen Spots zu finden. Werden einige Pixel diese Spots gefunden, andere nicht, ist Rauschen das Ergebnis. Mit Erhöhung der Samples steigt die Wahrscheinlichkeit alle Lichtquellen zu finden, aber auch die Renderzeit.

Mit einigen Tricks kann man dieses Rauschen reduzieren. Werden die hellen Spots unscharf (blur) gemacht, werden sie etwas größer und weniger konzentriert und sind damit besser zu finden und rauschärmer. Das ist zwar nicht exakt das gleiche Ergebnis, aber häufig nah genug an der erwünschten Wirkung. Unten ein Beispiel der Filter glossy und smooth light falloff.



Reflektionen

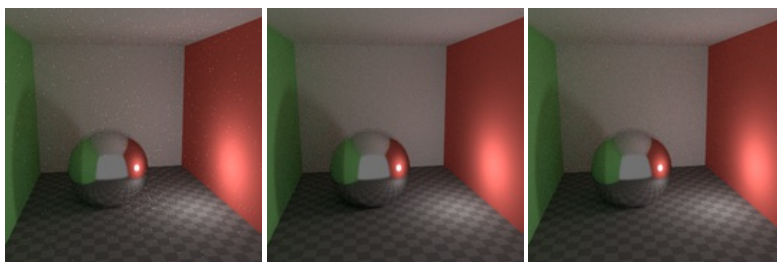
In Wirklichkeit wird das Licht durch seine hohe Geschwindigkeit unzählige Male hin- und herreflektiert. In der Praxis des Renderns verursachen viele Reflektionen aber erhöhtes Rauschen und es ist sinnvoll die Voreinstellung für Limited Global Illumination zu benutzen, da diese weniger Samples für die verschiedenen Shadertypen benutzt. Matte Oberflächen kommen üblicherweise mit wenigen Samples aus, während glänzende einige mehr brauchen und transparente Materialien, z.B. Glass, üblicherweise die meisten.



Ebenalls wichtig ist es, die Werte der shader colors, also die Helligkeit des Materials, nicht auf 1 oder nahe 1 zu setzen. Besser ist es die Werte auf maximal 0.8 zu begrenzen und das Licht heller zu machen. Zu hohe Werte tendieren zum Rauschen, da die Helligkeit, im Gegensatz zur Wirklichkeit, beim Reflektieren kaum abnimmt.

Kaustiken und Glossy Filter

Caustics are a well known source of noise causing fireflies. They happen because the renderer has difficulty finding specular highlights viewed through a soft glossy or diffuse reflection. There is a [No Caustics](#) option to disable glossy behind a diffuse reflection entirely. Many render engines will typically disable caustics by default.

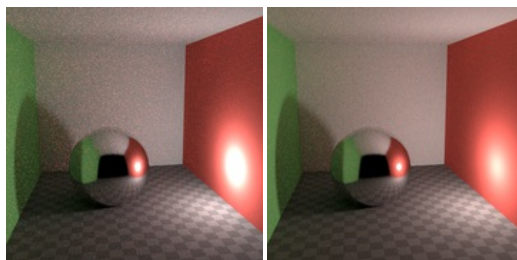


However using No Caustics will result in missing light, and it still does not cover the case where a sharp glossy reflection is viewed through a soft glossy reflection. There exists a [Filter Glossy](#) option to reduce the noise from such cases at the cost of accuracy. This will blur the sharp glossy reflection to make it easier to find, by increasing the shader Roughness.

The above images show default settings, no caustics, and filter glossy set to 1.0.

Lichtabfall

In reality light in a vacuum will always falloff at a rate of $1/(\text{distance}^2)$. However as distance goes to zero, this value goes to infinity and we can get very bright spots in the image. These are mostly a problem for indirect lighting, where the probability of hitting such a small but extremely bright spot is low and so happens only rarely. This is a typical recipe for fireflies.

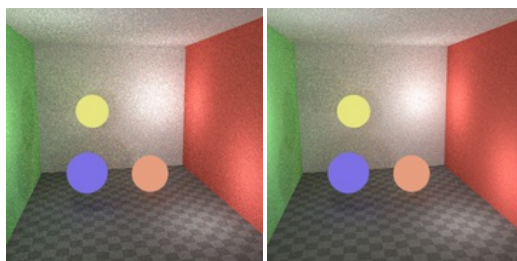


To reduce this problem, the [Light Falloff node](#) has a **Smooth factor**, that can be used to reduce the maximum intensity a light can contribute to a nearby surfaces. The images above show default falloff and smooth value 1.0.

Lampe als Beispiel

Materials with emission shaders can be configured to be [Sampled as a Lamp](#). This means that they will get rays sent directly towards them, rather than ending up there based on rays randomly bouncing around. For very bright mesh light sources this can reduce noise significantly. However when the emission is not particularly bright, this will take away samples from other brighter light sources for which it is important to find them this way.

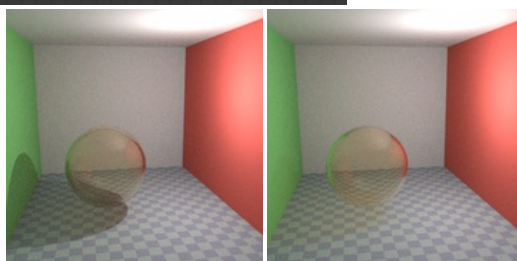
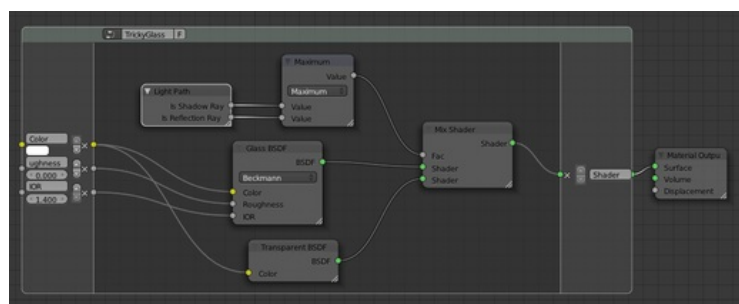
The optimal setting here is difficult to guess, it may be a matter of trial and error, but often it is clear that a somewhat glowing objects may be only contributing light locally, while a mesh light used as a lamp would need this option enabled. Here is an example where the emissive spheres contribute little to the lighting, and the image renders with slightly less noise by disabling Sample as Lamp on them.



The world background also has a [Sample as Lamp](#) option. This is mostly useful for environment maps that have small bright spots in them, rather than being smooth. This option will then in a preprocess determine the bright spots, and send light rays directly towards them. Again, enabling this option may take away samples from more important light sources if it is not needed.

Glas und Transparente Schatten

With caustics disabled glass will miss shadows, and with filter glossy they might be too soft. We can make a glass shader that will **use a Glass BSDF when viewed directly, and a Transparent BSDF when viewed indirectly**. The Transparent BSDF can be used for transparent shadows to find light sources straight through surfaces, and will give properly color shadow, but without the caustics. The Light Path node is used to determine when to use which of the two shaders.

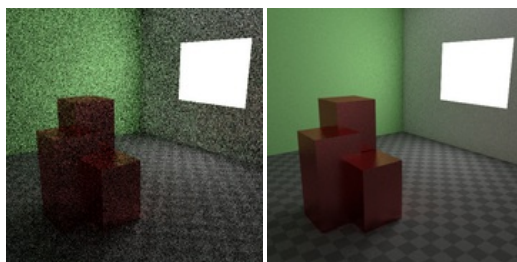


Above we can see the node setup used for the glass transparency trick, with on the left the render with too much shadow due to missing caustics, and on the right the render with the trick.

Fenster Lichter

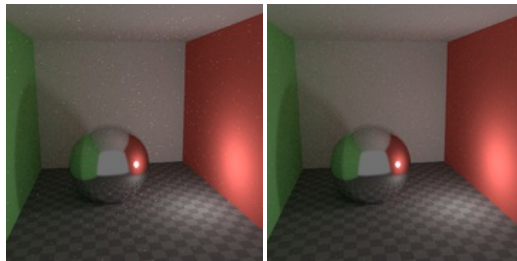
When rendering a daylight indoor scene and most of the light is coming in through a window or door opening, it is difficult for the integrator to find its way to them. We can replace with a plane with an emission shader, placed inside the window opening, so that the integrator knows in which direction to fire rays. For camera rays we can make this mesh light invisible, so that we can still look into the outside scene. This is done either by disabling camera ray visibility on the object, or by switching between glass and emission shaders in the material.

The two renders below have the same render time, with the second render using a mesh light positioned in the window.



Eingeklemmte Glühwürmchen

Ideally with all previous tricks fireflies would be eliminated, but they could still happen. For that **the intensity that any individual light ray sample will contribute to a pixel can be clamped** to a maximum value with the integrator [Clamp setting](#). If set too low this can give missing highlights in the image, which might be useful to preserve for camera effects such as bloom or glare.



GPU Rendering

Einführung

GPU Rendering ermöglicht es Ihre Grafikkarte für das Rendern anstelle ihrer CPU zu nutzen. Dies kann den Render Prozess beschleunigen, da moderne GPUs entwickelt werden eine große Anzahl an Zahlen zu verarbeiten. Auf der anderen Seite, weisen sie auch Grenzen für das Rendern komplexer Szenen, aufgrund eingeschränkten Speichers, und Problemen der Interaktivität auf, wenn die selbe Grafikkarte für das Anzeigen und Rendern genutzt wird.

Einrichtung

Um GPU Rendering zu aktivieren, gehen Sie auf Benutzereinstellungen, unter den Reiter System, und wählen dass Gerät zur Berechnung aus. Als nächstes, können Sie für jede Szene in den Render Einstellungen auswählen ob Sie mittels CPU oder GPU rendern lassen möchten.

CUDA

NVidia CUDA wird für GPU Rendering mit **NVidia Grafikkarten** unterstützt. Wir unterstützen Grafikkarten ab der GTX 2xx (Shader Model 1.3) Reihe aus, nichtsdestotrotz *wird empfohlen GTX 4xx oder GTX 5xx Karten* (Shader Model 2.x) zu nutzen, da für gewöhnlich nur mit diesen Karten eine gute Geschwindigkeit erzielt werden kann, frühe Karten sind oft langsamer als nur die CPU zu nutzen. Shader Model 1.3 Karten unterstützen auch nicht manche Cycles Funktionen, siehe unten.

Cycles benötigt aktuelle installierte Treiber, auf jedem Betriebssystem. Stellen Sie sicher dass die Blender Version die richtige für Ihr Betriebssystem ist, laden Sie 64 bit Blender für 64 Bit Betriebssysteme herunter.

[Liste von CUDA Karten mit Shader Model](#)

Ältere Karten

Für Mac und Linux, ist es möglich Kernel während der Laufzeit zu kompilieren, für Karten die offiziell nicht unterstützt werden. GeForce 8xxx, 9xxx Karten werden in der offiziellen Veröffentlichung nicht unterstützt, aber sie könnten durch das Aktivieren der [experimentellen Funktionen](#) funktionieren. Die [CUDA Toolkit Version 4.0](#) (64 bit Version) oder neuer muss dafür installiert sein. Andere Versionen können funktionieren, werden aber nicht unterstützt. Zum ersten Mal wenn der Render Prozess abgeschlossen ist, muss der Kernel für die entsprechende GPU Architektur kompiliert werden. Wegen der erhöhten Komplexität von Cycles im Gegensatz zu einem typischen GPU Kernel, kann der Kompilier-Vorgang 40 Sekunden bis zu einigen Minuten andauern, und auch über 2GB Speicher nutzen, abhängig vom Grafikmodell.

Fehlende Funktionen mit Shader Model 1.x

Aufgrund Grenzen seitens der Hardware, ist das Kompilieren eines Kernels mit allen aktivierten Funktionen für diese Karten nicht möglich. Momentan fehlen:

- Transparente Schatten
- Abtastung (Sample) als Lampe für Welt Texturen
- Belichtete Okklusion
- Render Durchläufe

OpenCL

Noch nicht fertig implementiert, als [experimentelles Feature](#) markiert.

Rendern mittels **OpenCL wird noch nicht ganz unterstützt**, aber es wird dran gearbeitet damit wir mehr Grafikkarten unterstützen können. Momentan wird nur einfaches Clay Rendering unterstützt, da unser ganzer Kernel nicht mit dem AMD OpenCL Kompilierer kompiliert werden kann.

Zum ersten Mal nach Abschluss des Render Vorgangs, muss der Kernel für Ihre GPU Architektur kompiliert werden.

Wegen der erhöhten Komplexität von Cycles im Gegensatz zu einem typischen GPU Kernel, kann der Kompilier-Vorgang 40 Sekunden bis zu einigen Minuten andauern, und auch über 2GB Speicher nutzen, abhängig vom Grafikmodell.

OpenCL Version 1.1 oder höher wird vorausgesetzt.

Oft gestellte Fragen (FAQ)

Warum reagiert Blender nicht während des Render Vorgangs?

Wenn eine Grafikkarte rendert, kann sie nicht die Benutzeroberfläche neu zeichnen, was Interaktivität unmöglich macht. Wir versuchen dieses Problem durch die Rückgabe der Kontrolle an die GPU so oft wie möglich zurückzugeben, aber eine komplett flüssige Interaktion kann nicht garantiert werden, speziell auf großen Bildschirmen. Dass ist eine Grenze der Grafikkarte für die keine klare Lösung existiert, aber wir können versuchen dass in der Zukunft zu verbessern.

Wenn möglich, ist es das Beste mehr als eine GPU einzurichten, eine zum Anzeigen und die andere(n) zum Rendern.

Warum rendert eine Szene auf der CPU, aber nicht auf der GPU?

Es können viele Gründe möglich sein, aber meistens ist es der, dass nicht genügend Speicher auf der Grafikkarte vorhanden ist. Wir

können momentan nur Szenen rendern die in den Grafikkarten Speicher passen, welcher für gewöhnlicher kleiner als der der CPU ist. Beachten Sie zum Beispiel dass 8000, 4000, 2000 und 1000 Bildtexturen 256MB, 64MB, 16MB und 4MB an Speicher einnehmen.

Wir haben die Absicht ein System hinzuzufügen dass Szenen unterstützt die größer als der GPU Speicher sind, aber dieses wird nicht all zubald hinzugefügt.

Kann Ich mehrere GPUs für den Render Vorgang nutzen?

Ja, gehen Sie zu Benutzereinstellungen > System > Computerberechnungen Panel, und konfigurieren Sie es nach Ihren Wünschen.

Würden mehrere GPUs den verfügbaren Speicher erhöhen?

Nein, jede GPU kann nur auf ihren eigenen Speicher zugreifen.

Was rendert schneller, NVidia oder AMD, CUDA oder OpenCL?

Momentan rendert NVidia mit CUDA schneller. Es gibt keinen wirklichen Grund warum dass so ist, wir nutzen keine CUDA spezifischen Funktionen, aber der Kompilierer scheint reifer zu sein, und kann besser größere Kernels unterstützen. An der OpenCL Unterstützung wird immer noch gearbeitet und sie ist noch nicht so optimiert, weil der gesamte Kernel bei uns noch nicht funktioniert hat.

Fehlermeldungen

unsupported GNU version! gcc 4.5 and up are not supported!

Auf Linux, abhängig von Ihrer GCC Version können Sie diese Fehlermeldung erhalten.

Falls dies der Fall ist löschen Sie die folgende Zeile in /usr/local/cuda/include/host_config.h

```
#error -- unsupported GNU version! gcc 4.5 and up are not supported!
```

CUDA Error: Invalid kernel image

Falls Sie diese Fehlermeldung auf Windows 64 bit erhalten, stellen Sie sicher die 64 bit Version von Windows zu nutzen, nicht die 32 bit Version.

CUDA Error: Out of memory

Dies bedeutet für gewöhnlich dass nicht genügend Speicher auf der GPU für das Speichern der Szene bereitsteht. Wir können momentan nur Szenen rendern die in den Grafikkarten Speicher passen, welcher für gewöhnlicher kleiner als der der CPU ist. Siehe oben für mehr Details.

The NVIDIA OpenGL driver lost connection with the display driver

... due to exceeding the Windows Time-Out limit and is unable to continue.

Wenn eine GPU für das Anzeigen und Rendern genutzt wird, hat Windows ein Limit an Zeit dass die GPU für Render Operationen nutzen darf. Wenn Sie eine teilweise große Szene haben, kann Cycles zu viel GPU Zeit in Anspruch nehmen. Das Reduzieren der Kachel-Größe im Leistungs Panel kann das Problem lindern, aber die einzig wirkliche Lösung ist seperate Grafikkarten für das Anzeigen und Rendern zu benutzen.

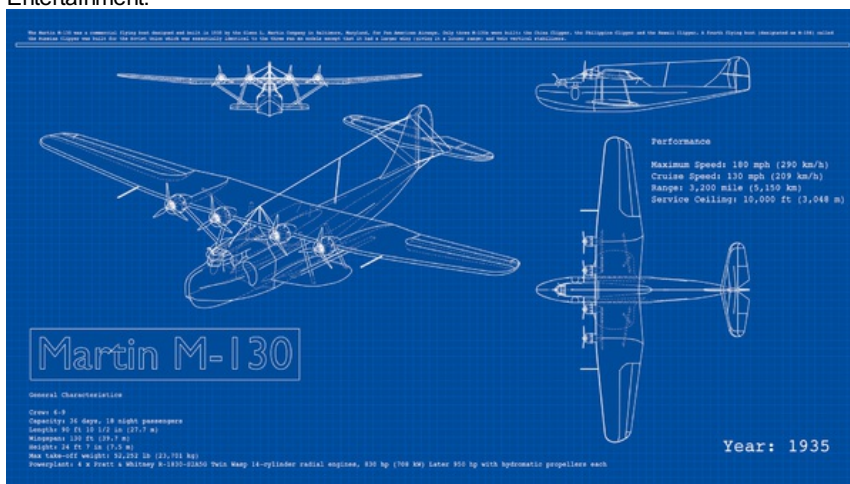
Was ist FreeStyle?

Freestyle ist eine Kanten- und linienbasierte non-fotorealistische (NPR) Rendering Engine für Blender. Sie basiert auf Mesh-Daten und Informationen von der Tiefe der Z-Achse um Linien auf ausgewählten Kantentypen zu zeichnen. Unterschiedliche Linientypen sind verfügbar, um artistisches ("von Hand gezeichnetes", "gemaltes", usw.) oder technisches (harte Linien) Aussehen zu realisieren.

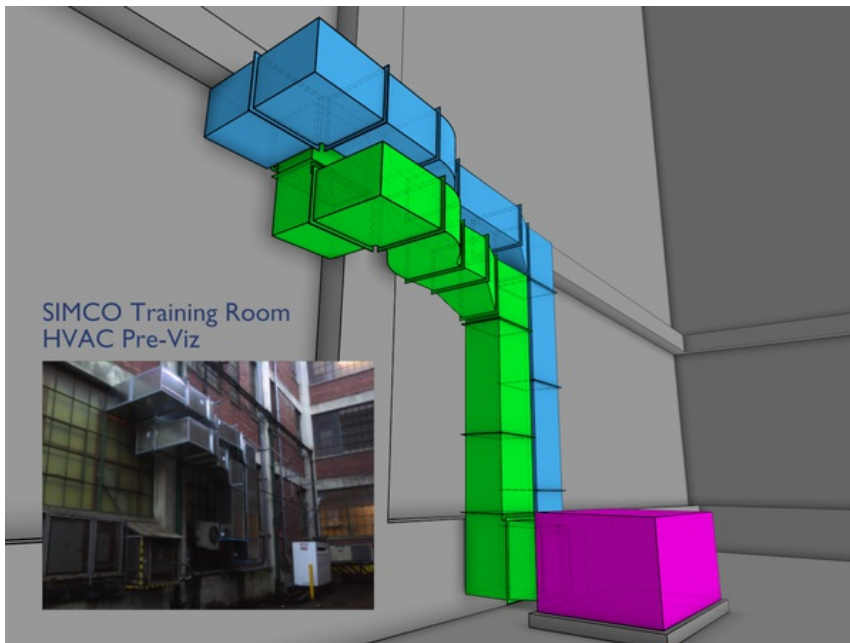
Es wird zwischen zwei Modi der Nutzung unterschieden - Einerseits [Python Scripting](#) und andererseits dem [Parameter Editor](#). Diese erlauben eine leistungsfähige Vielfalt von Linienstilen und Ergebnissen. Linienstile wie der japanische große Pinsel Line, Cartoon, Blueprint (deutsch: Blaupause), thickness-with-depth sind bereits als Skripte in Python verfügbar. Der Parameter Editor Modus erlaubt ein intuitives Verändern von außergewöhnlichen Funktionen, wie den gepunkteten Linien und ein einfaches Einrichten von mehreren Linientypen und Kantendefinitionen. Darauf aufbauend, ist mir der Einführung der Linienstil-Modifikatoren der Himmel die einzige Grenze!



Eine Cartoon Szene von [OHA Studio](#) (die .blend Datei). © Mechanimation Entertainment.



Blueprint Render von Martin M-130 aus dem Jahre 1935 von LightBWK. CC0.
ACHTUNG: GROßE DATEI! DESIGNED ALS STRESS TEST FÜR BLENDER, UND BLENDER AN DIE GRENZEN ZU BRINGEN, ABSTÜRZE SIND MÖGLICH. ([File:M-130Blueprint.zip](#))



HVAC Pre-Viz von Lee Posey. CC0 ([File:HVACPreViz.zip](#))



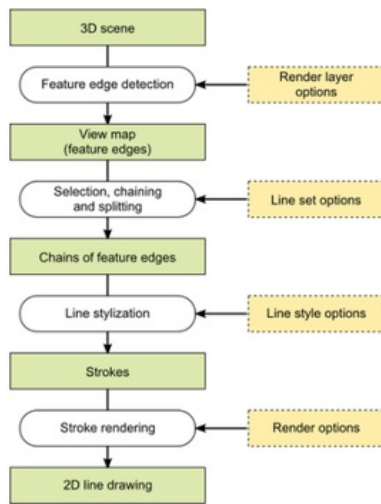
Küche von Vicente Carro. © AnigoAnimation

Mehr Kunstwerke sind verfügbar unter

http://wiki.blender.org/index.php/Dev:Ref/Release_Notes/2.67/FreeStyle#Freestyle_Artwork_Showcase

Das Gesamtbild

1. Freestyle kann gestartet werden über das Knöpfe Fenster → Render Tab → FreeStyle Panel, aktivieren der Checkbox. Bitte beachten Sie dass FreeStyle nur für die Blender Interne Rendering Engine verfügbar ist.
2. Die Einstellungen von Freestyle sind verfügbar im Kontext der Render Ebenen.
3. Einer Render Ebene kann nur eine Viewmap zugewiesen werden. Eine Viewmap beinhaltet die Erkennungseinstellungen für Kanten (Falten Winkel, mit Schaltern zur Ausmerzung, Face Glättung, und Material Begrenzungen, Bereichsradius und erweiterte Optionen für Kr Derivatives Epsilon).
4. Eine Viewmap kann mehrere Liniensets beinhalten.
5. Ein Linienset kontrolliert welche Linientypen und Auswahlen gerendert werden, basierend auf Linien aus Ihrer Szene.
6. Jedes Linienset nutzt einen Linienstil (welcher über mehrere Liniensets hinweg nutzbar ist).
7. Ein Linienstil beinhaltet die Infos für Freestyle, die für das Rendering von verknüpften Liniensets in Form von Farbe, Alpha, Dicke und anderen Aspekten notwendig sind.



Block Diagram von der Freestyle Viewmap und Prozessen

Bekannte Grenzen und Probleme

- FreeStyle is only available for Blender Internal renderer.
- Highly memory demanding: All mesh objects in a render layer are loaded at once.
- Only faced mesh objects are supported. The following kinds of meshes are ignored.
 - Mesh faces with wire materials.
 - Mesh faces with completely transparent materials.
- Transparent faces are treated as opaque faces.
- No edges at face intersection are detected yet.
- Layer masks do not work with Freestyle.
- Freestyle rendering results do not have any Z depth information
- Not working with panoramic camera.

Parameter Editor



Parameter Editor

Der Freestyle Parameter Editor bietet eine benutzerfreundliche Benutzeroberfläche zur Definition und Kontrolle von Liniensets und Linienshilen.

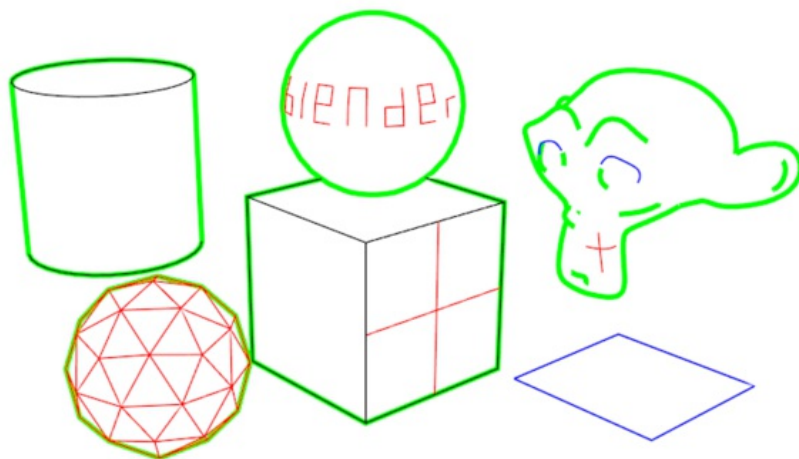
[Linesets](#) steuern welche der von Freestyle erkannten Kanten für das Rendering verwendet werden.

[Linestyle](#) kontrollieren wie die ausgewählten Kanten gerendert werden.

Eine Viewmap (daher eine Render-Ebene) kann mehrere Liniensets beinhalten, und jedes Liniensets ist verknüpft zu einem Linienshilen.

Linienset

Ein Linienset ... selects among the lines (edges) detected by Freestyle which ones will be rendered using its attached [line style](#), through various methods.



Examples of some basic edge types by LightBWK ([File:EdgeType.zip](#))

Auswahl nach Sichtbarkeit

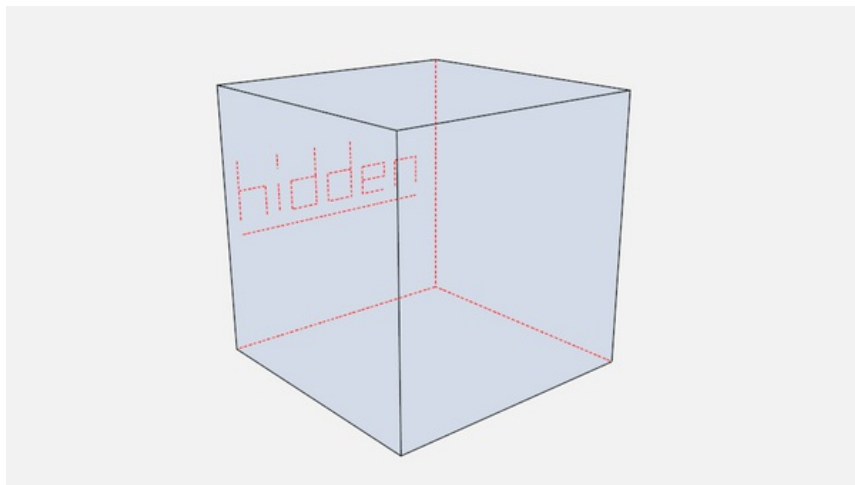
There are three choices for selecting edges by visibility.

Visible

Only lines occluded by no surfaces are rendered.

Hidden

Lines occluded by at least one surface are rendered.



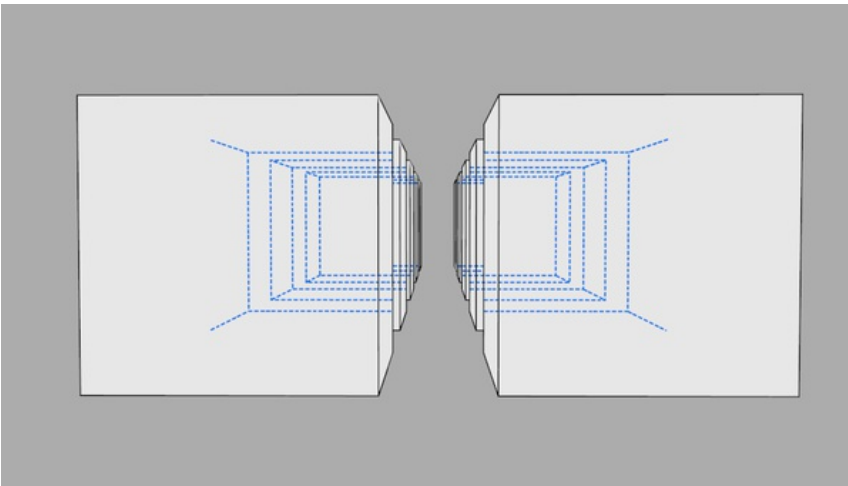
Proof of concept of visible and hidden edges by LightBWK
([File:HiddenCreaseEdgeMark.zip](#))

QI Range

QI stands for *Quantitative Invisibility*. Lines occluded by a number of surfaces into the given range are rendered.

Start and End

Only with QI Range, min/max number of occluding surfaces for a line to be rendered.



QI Range proof of concept demo, Start: 3, End: 7, by LightBWK ([File:QI-Range.zip](#))

Selection by Edge Types

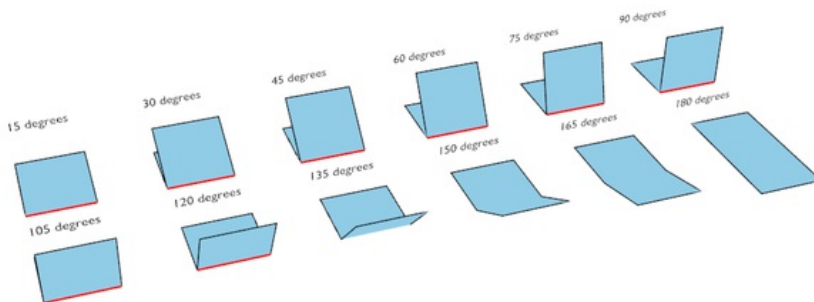
Edge types are the basic algorithms for the selection of lines from geometry. When using the parameter editor you have to choose at least one edge type in order to get a render output, but several edge types can be combined in one line set. Edge types can also be excluded from calculation by pressing the X next to them.

Silhouette

Draws silhouettes around your closed objects, it is often good for organic objects (like Suzanne & Sphere), and bad for sharp edges like a box. It can't render open mesh object like open cylinder and flat plane. The output is affected by the Kr Derivative Epsilon viewmap setting.

Crease

Shows only edges whose adjacent faces form an angle greater than the defined viewmap's Crease Angle.



Crease Angle proof of concept for 121° by LightBWK ([File:CreaseAngle.zip](#))

Border

Border is for open/unclosed edge mesh, an open cylinder has open edge at the top and bottom, a plane is open all around. Suzanne's eye socket is an open edge. All open edges will have lines rendered. Depends on mesh structure.

Edge Marks

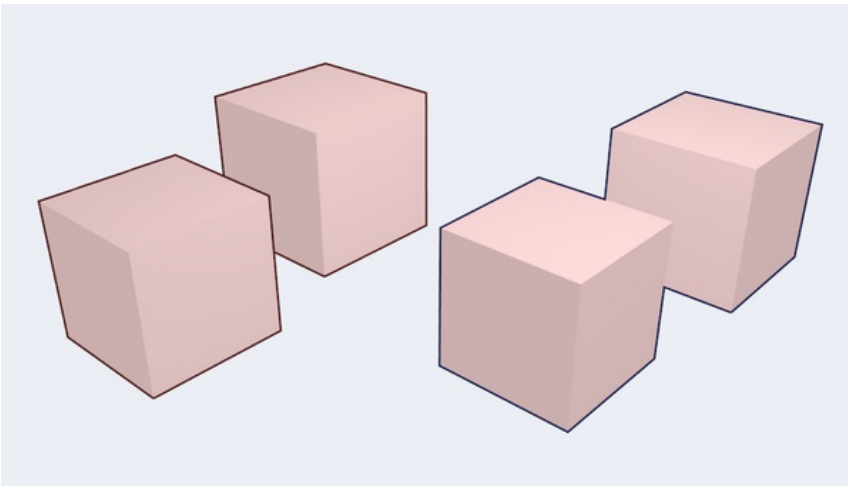
Renders marked edges. See [below](#) for details.

Contour

Draws the outer edges and inner open border.

External Contour

Draws the contour lines, but only on the outer edges.



Left pair: Contour; Right pair: External Contour

Suggestive Contour

Draws some lines which would form the contour of the mesh if the viewport was shifted. Depends on your viewmap settings for Kr Derivative Epsilon and Sphere Radius (further information: <File:Manual-2.6-Render-Freestyle-PrincetownLinestyle.pdf>).

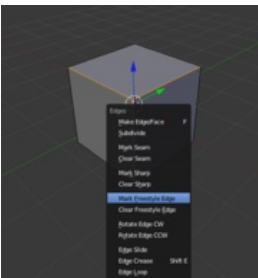
Material Boundary

Draws lines where two materials meet on the same object. Must be activated in the viewmap settings.

Ridge & Valley

Draws ridges and valleys. Depends on your Sphere Radius viewmap settings.

Edge Marks



Select and mark Freestyle edges.



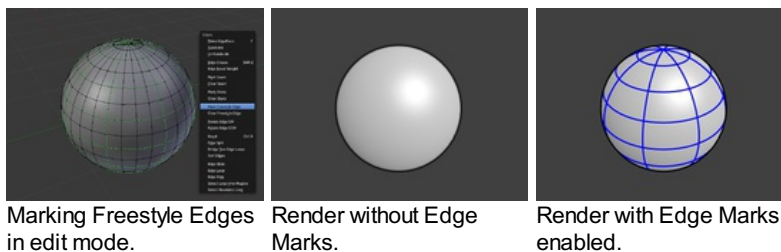
Edge Mark setting in the Line Sets tab.

In edit mode you can mark "Freestyle Edges" in the same manner you can mark "Seams" for UV unwrapping or "Sharp" for edge split. These marked edges are available to render when you select Edge Mark.

This is done as follows:

- Select your mesh and tab into Edit mode.
- Select the edges you want to be marked.
- Press **Ctrl+E** and select Mark Freestyle Edge.

Edge marks are useful when you want to draw lines along particular mesh edges. The examples below explains the use of edge marks.



Marking Freestyle Edges in edit mode.

Render without Edge Marks.

Render with Edge Marks enabled.

The image on the left shows a sphere in Edit mode. The green lines are the edge marks. On the right you see a render without edge marks enabled.

With edge marks enabled, the previously marked lines are always rendered. You can see the black contour lines and the blue lines that are made with edge marks.

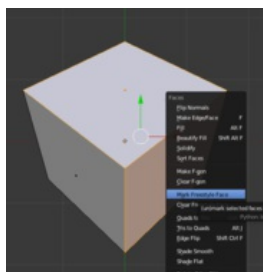
What edge marks are good for?

1. When you need to render marks on almost flat plane, when other edge types can't detect any line.
2. When you want full control of edge rendering. Often for edges of squarish shapes.
3. Mark the whole base mesh to be rendered for base mesh preview.

What edge marks are not for?

1. Round outer edges (use instead Contour/External Contour/Silhouette).

Selection by Face Marks



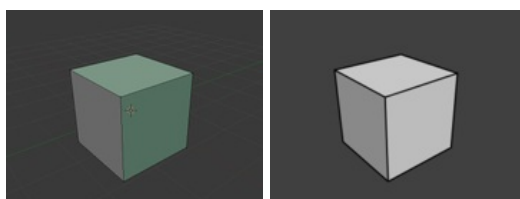
Mark Freestyle Faces.

To set a face mark:

- Select a mesh and tab into Edit mode.
- Select the faces you want to be marked.
- Press CtrlF and select Mark Freestyle Face.

Face marks are useful for removing lines from certain areas of a mesh.

In this example two faces of the default cube are marked like the image on the left. On the right is a render without face marks activated.



Marked Faces.

Render Output.



Face mark options.

The line selection can be controlled via inclusion and faces options:

Inclusive/Exclusive

Whether to include or exclude edges matching defined face mark condition from the line set.

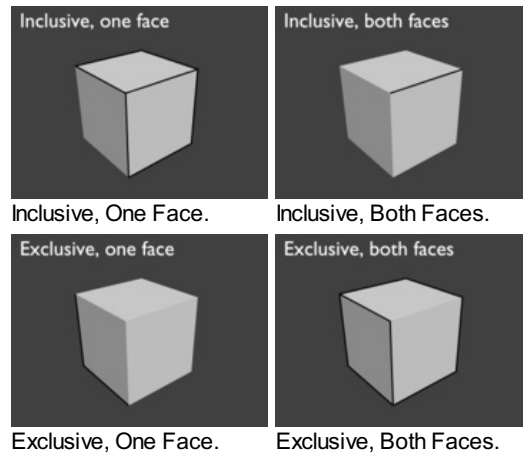
One Face

(De)select all edges which have one of both neighbor faces marked.

Both Faces

(De)select all edges which have both of their neighbor faces marked.

The image below shows the resulting combinations.



Selection by Group

You can include or exclude objects for line calculation, based on their belonging to a group.

Group

The name of the object group to use.

Inclusive/Exclusive

Whether to include or exclude lines from those objects in this line set.

Selection by Image Border

If enabled, Freestyle only takes geometry within the image border into consideration for line calculation. This reduces render times but increases continuity problems when geometry is moved out of and into camera view.

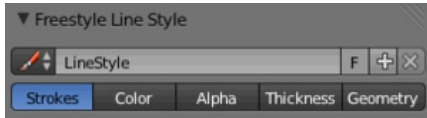
Linenstil & Modifikatoren

In Freestyle, the line style settings define the appearance of a line set by five main aspects: [stroke](#), [color](#), [alpha](#), [thickness](#) and [geometry](#). All those allow you to get many different styles of renders (technical draw, rough sketch, cartoon, oriental calligraphy, etc.).

You can create as much line styles as you wish, and reuse a given line style for several line sets by selecting it from the dropdown menu next to its name.

Length Unit

Unless otherwise specified, all length in line style settings are in pixels (either relative or absolute, as specified in the [core options](#)).



Line Style UI



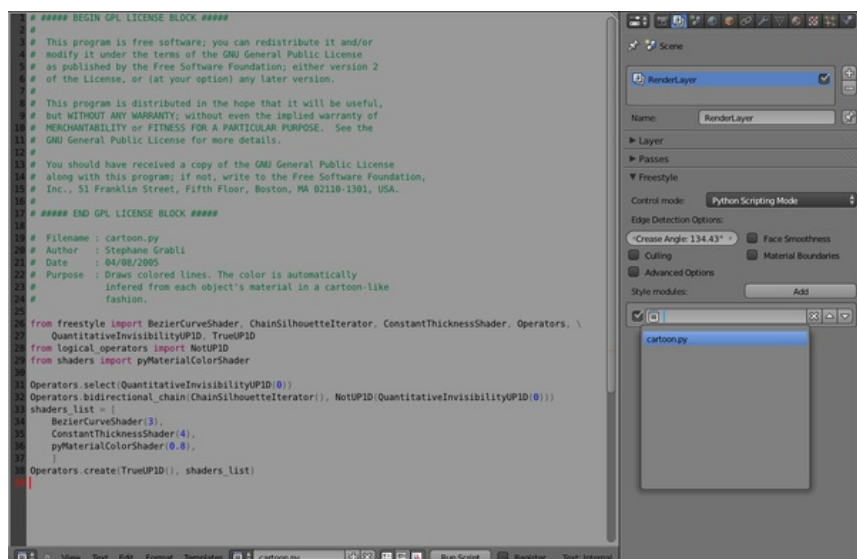
Line Style demo [File:LineStyle.zip](#)

Python Scripting Modus

Der Python Scripting Modus bietet eine komplette Programmierbarkeit der Gestaltung von Linien. In diesem Kontrollmodus, werden alle Style Operationen als Python Skripte in der Freestyle Terminologie referenziert. Die Eingabe zu einem Style Modul ist eine Viewmap (z.B., eine Zusammenstellung von erkannten Funktionskanten), und die Ausgabe bildet ein Zusammenstellung aus gestalteten Linien.

Ein Style Modul setzt sich aus ... successive calls of five basic operators: selection, chaining, splitting, sorting and stroke creation. The selection operator identifies a subset of input feature edges based on one or more user-defined selection conditions (predicates). The selected edges are processed with the chaining, splitting and sorting operators to build chains of feature edges. These operators are also controlled by user-supplied predicates and functions in order to determine how to transform the feature edges into chains. Finally, the chains are transformed into stylized strokes by the stroke creation operator, which takes a list of user-defined stroke shaders.

Python style modules are stored within .blend files as text datablocks. External style module files first need to be loaded in the Text Editor window. Then the pull-down menu within an entry of the style module stack allows you to select a module from the list of loaded style modules.



A screen capture of a style module (cartoon.py) loaded in the Text Editor window (left), as well as Freestyle options in the Python Scripting mode in the Render Layers buttons (right)

Freestyle for Blender comes with a number of Python style modules that can serve as a starting point of your own style module writing. See also the section of the Freestyle Python API in the Blender Python API reference manual for the full detail of style module constructs.



By T.K. using the Python Scripting mode
([File:Turning Pages.zip](#), CC0)



By T.K. using the Python Scripting mode
([File:Lily Broken Topology.zip](#), CC0)

Writing Style Modules

A style module is a piece of code responsible for the stylization of Freestyle line drawing. The input of a style module is a set of feature edges called view map (ViewMap). The output is a set of stylized lines also referred to as strokes. A style module is structured as a pipeline of operations that allow for building strokes from the input edges within the view map. There are five kinds of operations (corresponding operator functions in parentheses):

- Selection (Operators.select())
- Chaining (Operators.chain(), Operators.bidirectional_chain())
- Splitting (Operators.sequential_split(), Operators.recursive_split())
- Sorting (Operators.sort())
- Stroke creation (Operators.create())

The input view map is populated with a set of ViewEdge objects. The selection operation is used to pick up ViewEdges of interest to artists based on user-defined selection conditions (predicates). Chaining operations take the subset of ViewEdges and build Chains by concatenating ViewEdges according to user-defined predicates and functions. The Chains can be further refined by splitting them

into smaller pieces (e.g., at points where edges make an acute turn) and selecting a fraction of them (e.g., to keep only those longer than a length threshold). The sorting operation is used to arrange the stacking order of chains to draw one line on top of another. The chains are finally transformed into stylized strokes by the stroke creation operation applying a series of stroke shaders to individual chains.

ViewEdges, Chains and Strokes are generically referred to as one-dimensional (1D) elements. A 1D element is a polyline that is a series of connected straight lines. Vertices of 1D elements are called 0D elements in general.

All the operators act on a set of active 1D elements. The initial active set is the set of ViewEdges in the input view map. The active set is updated by the operators.

Selection

The selection operator goes through every element of the active set and keeps only the ones satisfying a certain predicate. The `Operators.select()` method takes as the argument a unary predicate that works on any `Interface1D` that represents a 1D element. For example:

```
Operators.select(QuantitativeInvisibilityUP1D(0))
```

This selection operation uses the `QuantitativeInvisibilityUP1D` predicate to select only the visible `ViewEdge` (more precisely, those whose quantitative invisibility is equal to 0). The selection operator is intended to selectively apply the style to a fraction of the active 1D elements.

It is noted that `QuantitativeInvisibilityUP1D` is a class implementing the predicate that tests line visibility, and the `Operators.select()` method takes an instance of the predicate class as argument. The testing of the predicate for a given 1D element is actually done by calling the predicate instance, that is, by invoking the `__call__` method of the predicate class. In other words, the `Operators.select()` method takes as argument a functor which in turn takes an `Interface0D` object as argument. The Freestyle Python API employs functors extensively to implement predicates, as well as functions.

Chaining

The chaining operators act on the set of active `ViewEdge` objects and determine the topology of the future strokes. The idea is to implement an iterator to traverse the `ViewMap` graph by marching along `ViewEdges`. The iterator defines a chaining rule that determines the next `ViewEdge` to follow at a given vertex (see `ViewEdgeIterator`). Several such iterators are provided as part of the Freestyle Python API (see `ChainPredicateIterator` and `ChainSilhouetteIterator`). Custom iterators can be defined by inheriting the `ViewEdgeIterator` class. The chaining operator also takes as argument a `UnaryPredicate` working on `Interface1D` as a stopping criterion. The chaining stops when the iterator has reached a `ViewEdge` satisfying this predicate during the march along the graph.

Chaining can be either unidirectional (`Operators::chain()`) or bidirectional (`Operators::bidirectional_chain()`). In the latter case, the chaining will propagate in the two directions from the starting edge.

The following is a code example of bidirectional chaining:

```
Operators.bidirectional_chain(ChainSilhouetteIterator(),
                             NotUP1D(QuantitativeInvisibilityUP1D(0)))
```

The chaining operator uses the `ChainSilhouetteIterator` as the chaining rule and stops chaining as soon as the iterator has come to an invisible `ViewEdge`.

The chaining operators process the set of active `ViewEdge` objects in order. The active `ViewEdges` can be previously sorted using the `Operators::sort()` method (see below). It starts a chain with the first `ViewEdge` of the active set. All `ViewEdges` that have already been involved in the chaining process are marked (in the case of the example above, the time stamp of each `ViewEdge` is modified by default), in order not to process the same `ViewEdge` twice. Once the chaining reaches a `ViewEdge` that satisfies the stopping predicate, the chain is terminated. Then a new chain is started from the first unmarked `ViewEdge` in the active set. This operation is repeated until the last unmarked `ViewEdge` of the active set was processed. At the end of the chaining operation, the active set is set to the Chains that have just been constructed.

Splitting

The splitting operation is used to refine the topology of each Chain. Splitting is performed either sequentially or recursively. Sequential splitting (`Operators::sequentialSplit()`) in its basic form, parses the Chain at a given arbitrary resolution and evaluates a unary predicate (working on 0D elements) at each point along the Chain. Every time the predicate is satisfied, the chain is split into two chains. At the end of the sequential split operation, the active set of chains is set to the new chains.

```
Operators.sequentialSplit(TrueUP0D(), 2)
```

In this example, the chain is split every 2 units. A more elaborated version uses two predicates instead of one: One to determine the starting point of the new chain and the other to determine its ending point. This second version can lead to a set of Chains that are disjoint or that overlap if the two predicates are different. (see `Operators::sequentialSplit()` for more details).

Recursive splitting (`Operators::recursiveSplit()`) evaluates a function on the 0D elements along the Chain at a given resolution and find the point that gives the maximum value for the function. The Chain is then split into two at that point. This process is recursively repeated on each of the two new Chains, until the input Chain satisfies a user-specified stopping condition.

```
func = Curvature2DAngleF0D()
Operators.recursive_split(func, NotUP1D(HigherLengthUP1D(5)), 5)
```

In the code example above, the Chains are recursively split at points of the highest 2D curvature. The curvature is evaluated at points along the Chain at a resolution of 5 units. Chains shorter than 5 units won't be split anymore.

Sorting

The sorting operator (`Operators::sort()`) arranges the stacking order of active 1D elements. It takes as argument a binary predicate used as a “smaller than” operator to order two 1D elements.

```
Operators.sort (Length2DBP1D())
```

In this code example, the sorting uses the `Length2DBP1D` binary predicate to sort the `Interface1D` objects in the ascending order in terms of 2D length.

The sorting is particularly useful when combined with causal density. Indeed, the causal density evaluates the density of the resulting image as it is modified. If we wish to use such a tool to decide to remove strokes whenever the local density is too high, it is important to control the order in which the strokes are drawn. In this case, we would use the sorting operator to insure that the most “important” lines are drawn first.

Stroke creation

Finally, the stroke creation operator (`Operators::create()`) takes the active set of Chains as input and build Strokes. The operator takes two arguments. The first is a unary predicate that works on `Interface1D` that is designed to make a last selection on the set of chains. A Chain that doesn't satisfy the condition won't lead to a Stroke. The second input is a list of Shaders that will be responsible for the shading of each built stroke.

```
shaders_list = [
    SamplingShader(5.0),
    ConstantThicknessShader(2),
    ConstantColorShader(0.2,0.2,0.2,1),
]
Operators.create(DensityUP1D(8,0.1, IntegrationType.MEAN), shaders_list)
```

In this example, the `DensityUP1D` predicate is used to remove all Chains whose mean density is higher than 0.1. Each chain is transformed into a stroke by resampling it so as to have a point every 5 units and assigning to it a constant thickness of 2 units and a dark gray constant color.

User control on the pipeline definition

Style module writing offers different types of user control, even though individual style modules have a fixed pipeline structure. One is the sequencing of different pipeline control structures, and another is through the definition of functor objects that are passed as argument all along the pipeline.

Different pipeline control structures can be defined by sequencing the selection, chaining, splitting, and sorting operations. The stroke creation is always the last operation that concludes a style module.

Predicates, functions, chaining iterators, and stroke shaders can be defined by inheriting base classes and overriding appropriate methods. See the reference manual entries of the following base classes for more information on the user-scriptable constructs.

- `UnaryPredicate0D`
- `UnaryPredicate1D`
- `BinaryPredicate0D`
- `BinaryPredicate1D`
- `UnaryFunction0DDouble`
- `UnaryFunction0DEdgeNature`
- `UnaryFunction0DFloat`
- `UnaryFunction0DId`
- `UnaryFunction0DMaterial`
- `UnaryFunction0DUnsigned`
- `UnaryFunction0DVec2f`
- `UnaryFunction0DVec3f`
- `UnaryFunction0DVectorViewShape`
- `UnaryFunction0DViewShape`
- `UnaryFunction1DDouble`
- `UnaryFunction1DEdgeNature`
- `UnaryFunction1DFloat`
- `UnaryFunction1DUnsigned`
- `UnaryFunction1DVec2f`
- `UnaryFunction1DVec3f`
- `UnaryFunction1DVectorViewShape`
- `UnaryFunction1DVoid`
- `ViewEdgeIterator`
- `StrokeShader`

Kompositions Nodes

Kompositions Nodes erlauben das zusammensetzen und verändern von Bildern oder Filmen. Es können Elemente auf Basis fast aller ihrer Eigenschaften in ihrem Aussehen verändert werden dazu gehören: Farbe, Ebene, Geschwindigkeit, Lichtabstrahlung und einige mehr. Das erlaubt das Hinzufügen von Leuchteffekten, das Verschwimmen von Bewegungen mit Motion Blur, also dem im Film auftretenden Verschwimmen von Elementen aufgrund von Geschwindigkeit und viele weitere Interessante Effekte die nichts mit den Modellen oder der Szene direkt zu tun haben, aber nötig sind um einen gewissen Effekt zu erzielen.

Node Konzepte

Nodes

"Nodes" sind individuelle Blöcke die gewisse Funktionen haben. Konzeptuell gibt es drei Arten von Nodes:

- **Eingabe Nodes**

Diese Nodes erzeugen oder importieren Information die dann von anderen Nodes verwendet wird und haben keinen Input vom Nodeeditor.

Beispiele sind: Bilder, Bildsequenzen oder Filme, Render Ebenen, Werte wie Farben oder einfach Zahlenwerte.

- **Verarbeitende Nodes:**

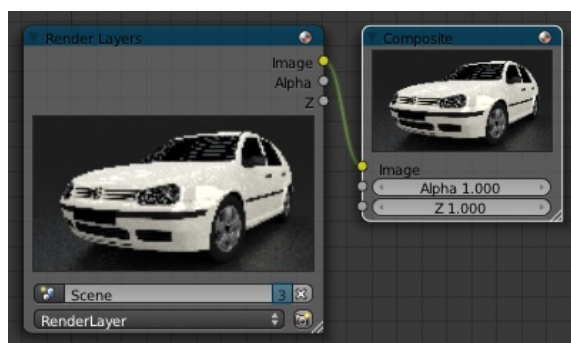
Diese Nodes leisten die eigentliche Arbeit. Sie filtern oder verändern Werte aus den Input Nodes.

Beispiele sind: Mathematische Nodes, RGB Kurven und Node für Verschwimmeneffekte.

- **Ausgabe Nodes:**

Diese Nodes erzeugen aus den von den Verarbeitenden Nodes erzeugten Informationen ein Ergebnis, also ein entsprechend veränderte Bilder oder Filme.

Beispiele sind: *Kompositions* Nodes (die das eigentliche Ergebnis erzeugen), *Anzeiger* (die die Ausgabe in Zwischenschritten darstellen) und Dateiausgabe nodes.



Nudeln

Die Idee hinter Nudeln ist das beliebig komplexe Veränderungen an Bilder vorgenommen werden können. Nudeln verbinden die Kompositionsnodes miteinander und übertragen die jeweiligen Elemente. Rechts ist eine Nudel zu sehen die die Nodes "Render Layers" und "Composite" verbindet und das Bild von der "Render Layers"node zur "Composite"node überträgt.

Node Gruppen

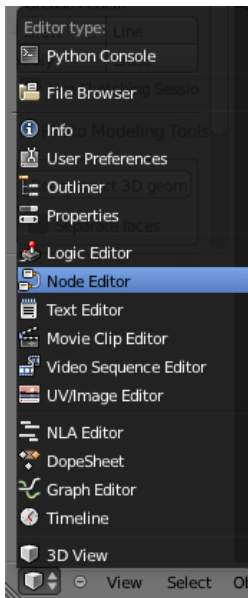
Es können Nodegruppen erzeugt werden die dann wie eine einzelne Node funktionieren und wie Objektgruppen aus anderen .blend Dateien importiert werden können.

Benutzen von Nodes

Im [Nodeeditor](#) muss zu allererst ausgewählt werden wofür der Nodeeditor verwendet werden soll. Es ist möglich den Nodeeditor für Materialien und für Nachbearbeitung zu benutzen. Es muss das entsprechende Icon aktiviert werden.



Nodeeditor Kopfzeile mit den Kompositionsnodes aktiviert.



Wähle das Nodeeditor
Fenster

Das Aktivieren der Nodes in einem entsprechenden Modus erzeugt standardmäßig einen Eingabe- und einen Ausgabeknoten, die ansich schon funktionieren, nur verändern sie noch nichts.

Jetzt müssen die entsprechenden Nodes hinzugefügt und mit den Knoten eingebunden werden um Effekte zu erzielen.

Beispiele

Es kann fast alles mit Nodes gemacht werden:

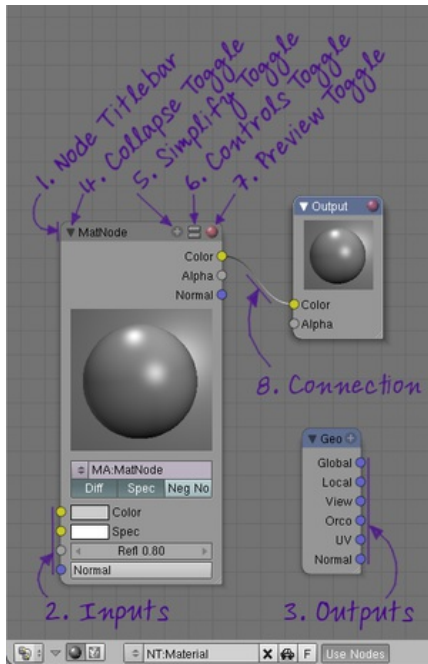
- Blue Screenverwendung oder ein bereits fertiges Bild können ebenso benutzt werden wie gerade erst in Blender erzeugte Bilder. Das ist ein wesentlicher Teil bei der Kombination von durch [Motion tracking](#) erzeugten Filmen in Blender und dem Original.
- Das verändern der Stimmung eines Bildes durch hinzufügen einer Verschiebung des Farbspektrums in verschiedene Farben.
- Flashbacks durch weiche Ränder oder Filmkörnung oder falsche Fehler im "Film".
- Kontraste können für Schlüsselereignisse erhöht werden.

und sehr viel mehr.

[Doc:DE/2.6/Reference/Nodes/Node Editor](#)

Node Kontrollen

Diese Seite erklärt wie man einzelne Nodes einstellt und kontrolliert.



Nodes main controlsHaupt node Kontrollen

Ansichtsoptionen

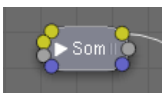
Kollabieren von Nodes

Kollabieren

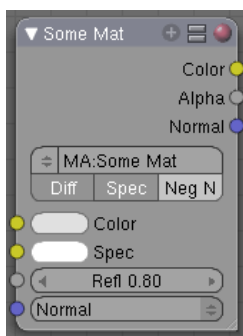
Links oben im Kopf der Node gibt es einen Knopf zum Kollabieren der Node. Sie erhält alle Verbindungen und Einstellungen, aber nimmt dann nicht mehr soviel Platz ein. Das kann helfen um bei komplizierten Nodenetzwerken den Überblick zu behalten.

Vorschaubildschalter ()

Der runde Knopf in der rechten oberen Ecke der der Node schaltet das Vorschaubild in der Node ein oder aus, was die Node verkleinert aber die Werte nicht versteckt wie das kollabieren.



Kollabierknopf



Runder Vorschaupfopf

Größenveränderung der Node

Falls die Nodes zu klein erscheinen können sie am Rand wie normale Fenster durch klicken, halten und ziehen der Maus vergrößert werden.

Bearbeitungseinstellungen

Sockel



Node

Socket.

Socket sind die Verbindungsschnittstellen von Nodes über sie werden Informationen von einem Node in einen anderen Übertragen.

Sockelfarben

Zur Übersichtlichkeit gibt es eine Farbkodierung für verschiedene Sockeltypen:

-  *blaue Sockel*

verarbeiten Vektorenwerte

-  *gelbe Sockel*

verarbeiten Farbwerte

-  *graue Sockel*

verarbeiten einzelne Werte oder eine "Wertekarte", die normalerweise in Grau dargestellt wird. Falls ein Eingabesockel eine Wertekarte als Eingabe erwartet und nur ein einzelner Wert eingegeben wird, werden alle Werte der Karte auf diesen Wert gesetzt.

Es können nur Nodes der gleichen Farbe sinnvoll verbunden werden, falls keine Umrechnungsnode verwendet wird.

Eingabesockel

Eingabesockel befinden sich an der linken Seite der Node und verwalten das Eingeben von Werten in den Node.

Ausgabesockel

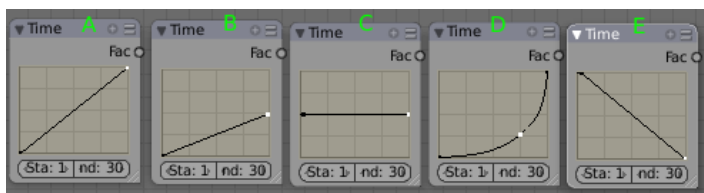
Ausgabesockel befinden sich an der rechten Seite der Node und verwalten das Ausgeben von Werten aus dem Node.

Werte

Wenn Werte nicht von anderen Nodes übernommen werden oder in Fällen wo das gar nicht möglich ist, sind viele Werte in Nodes auch direkt einstellbar. Dazu muss einfach auf den Wert oder das Farbfeld geklickt und der gewünschte Wert eingestellt werden.

Kurven

Einige Nodes haben ein Kurve als Einstellelement das den Eingabewert entsprechend der Kurve verändert. Die Kurve kann durch verschieben oder erzeugen der Signifikanten Punkte der Kurve verändert werden. Hier sind einige Beispiele:



Verändern einer Kurve in einer Node

Standardmäßig sind Kurven linear, was durch entsprechendes Verschieben der Kontrollpunkte geändert werden kann. Eine Veränderung der Kurve verändert die Erzeugung der Ausgabe folgendermaßen: Die Eingabe wird als X Wert interpretiert, der einem Y Wert durch die Kurve zugeordnet wird. Dieser Y Wert ist dann die Ausgabe.

RGB Kurven

Verschiedene Kurven können in einem Fenster verändert werden. Die RGB Kurve ist ein gutes Beispiel dafür, sie hat verschiedene Knöpfe mit denen der gewünschte Farbkanal verändert werden kann.

Bearbeitung der Ansicht

Standardmäßig ist die Ansicht auf ein Gebiet zwischen 0 und 1 Begrenzt. Diese Begrenzung kann jedoch durch klicken des # ähnlichen Symbols ausgeschaltet werden.

Spezial Werkzeuge

Das Schraubenschlüsselsymbol enthält Spezialwerkzeuge unter anderem zum zurücksetzen der Ansicht oder der Kurve.

Page status ([reviewing guidelines](#))

Text Diese Seite hat keinen Inhalte der nicht bereits in dieser (im Moment noch englischen) [Benutzeroberflächenstandards](#) Seite abgedeckt wird.

Proposed fixes: removal

Node Gruppen

Alle Nodes können gruppiert werden um eine verbesserte Übersichtlichkeit zu schaffen. Außerdem erlauben Nodegruppen einen export oder import nach oder aus .blend Dateien zum wiederverwenden von Nodegruppen.

Konzeptuell erstellt das Gruppieren von Nodes eine neue Node die neue Eingabesockel und Ausgabesockel haben und auf diese auf die in der Nodegruppe spezifizierte Weise reagieren.

Das Anhängen und Verlinken von Nodegruppen funktioniert genauso wie das Anhängen und Verlinken von [Objektgruppen](#).

Nodegruppen

Panel: [Nodeeditor](#)

Menu: ⇧ ShiftA → Gruppen

Eine Nodegruppe wird per Auswahl der Gewünschten Nodes und aktivieren der Gruppe erstellen Funktion mit StrgG oder der Menüfunktion.



Ein- und Ausgabe

Die zentrale Idee hinter Nodes ist die Wiederverwendbarkeit. Es macht also häufig keinen Sinn Ein- oder Ausgabeknoten in einer Gruppe zu verwenden.

Bearbeiten von Nodegruppen

Nodes haben einen eigenen Bearbeitungsmodus der mit ⇐ Tab erreicht und verlassen wird. Die Bearbeitung des Nodenetzwerks in der Gruppe funktioniert wie im Nodeeditor.

Gruppierungen auflösen

AltG löst eine Zusammenfassung in eine Gruppe wieder auf.

Typen von Kompositionsnodes

Hier sind alle verschiedenen Nodetypen nach Eigenschaft sortiert die im Nodeeditor verwendet werden können aufgeführt:

- [Eingabe](#) - Fügt dem Nodenetzwerk Information wie einen Wert oder ein Bild hinzu.
- [Ausgabe](#) - Zeigt das Ergebnis an oder gibt es aus.
- [Farbe](#) - Hier sind alle Farbmanipulationsnodes zu finden.
- [Vektor](#) - Hier sind alle Nodes zu finden die Vektorwerte verändern.
- [Filter](#) - Hier sind alle Filternodes zu finden.
- [Konverter](#) - Hier sind alle Konverter zu finden die die verschiedenen Datentypen ineinander umwandeln kann.
- [Masken](#) - Erzeugt Masken um die Veränderung des Bildes auf einen Ausschnitt zu begrenzen.
- [Verzerrung](#) - Hier sind alle Nodes zusammengefasst die das Bild verzerren können.
- [Gruppen](#) - Benutzerdefinierte Gruppen.

Kompositions Eingabe Nodes

Eingabe Nodes erzeugen oder importieren Information in den Nodeeditor.

Zum Beispiel:

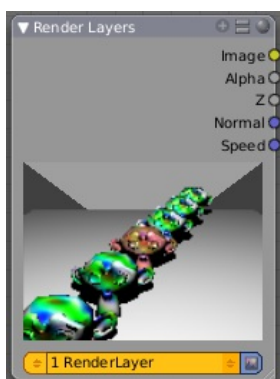
- Das gerade gerenderte Bild
- eine bereits existierende Bilddatei
- ein Filmclip
- eine Farbe

Diese Nodes erzeugen Informationen die die anderen Nodes benutzen. Offensichtlich brauchen sie keine Eingabesockel.

Render Ebenen Node

Panel: [Nodeeditor](#) → [Kompositionsnodes](#)

Menu: ⇧ ShiftA → [Input](#) → Render Ebenen



Die Render Ebenen Node

Dieser Node ist der Anfang um ein Bild in den Nodeeditor zu laden.

Er speist ein Bild aus einer Szene aus der .blend datei ein. Es muss eine Szene und eine aktive Renderebene aus der gelben Auswahlliste am unteren Ende der Node ausgewählt werden. Blender benutzt die aktive Kamera für diese Szene um ein Bild der Objekte auf dieser [Renderebene](#) zu erzeugen.

Das Bild wird zusammen mit den folgenden Daten in das Nodenetzwerk geladen:

- Alpha maske

Abhängig von den eingeschalteten Pässen der Renderebene können andere Sockel verfügbar sein. Standardmäßig ist Z eingeschaltet:

- Z Tiefe (wie weit ein Pixel von der Kamera entfernt ist)

Das Beispiel zeigt zwei weitere Eingeschaltete Kanäle:

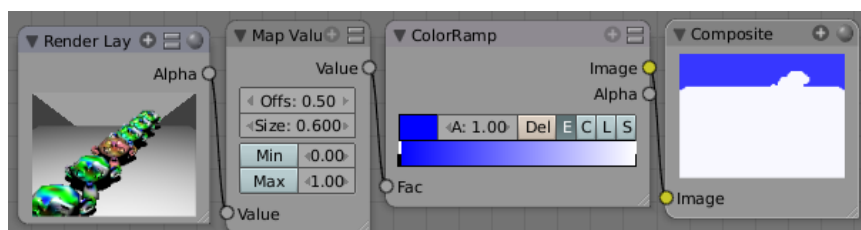
- Normalenvektorset (bestimmt wie das Licht von den Oberflächen reflektiert wird)
- Geschwindigkeitsvektorset (Wie schnell sich ein Objekt zum Zeitpunkt des Bilds bewegt)

Der Re-render (erneut Render) Knopf (kleines Landschaftsicon rechts vom Name der Renderebene kann dazu benutzt werden das Bild und damit das Netzwerk zu aktualisieren.

Eine .blend Datei enthält mehrere Szenen. Der Renderebenen Node kann die Szeneninformation aus jeder verfügbaren Szene mit Auswahl links auswählen, allerdings ist das die Information die vor dem Rendern und der Nachbearbeitung verfügbar ist. Falls es nötig sein sollte die zusammengestellte Information aus der Szene zu benutzen muss diese Szene in ein OpenEXR frameset gerendert werden das als Zwischenspeicher dient und es sollte dann der Bildimportnode benutzt werden.

Benutzung des Alphasockels

Der Alpha Sockel ist fundamental beim Übereinanderlegen von Bildern und das "Durscheinen" von unteren Bildern. In einer Blenderszene schweben die Objekte im virtuellen Raum. Obwohl einige vor anderen sind (Z Tiefe) gibt es keinen richtigen Hintergrund. Die Welteinstellungen erzeugen zwar die Illusion eines Horizonts aber es ist trotzdem nur eine Illusion. Außerdem sind einige Objekte halbtransparent; das wird ALphawert genannt. Ein halbtransparentes Objekt erlaubt es Licht (und Hintergrund Bildern) zur Kamera durchzuscheinen. Ein Nebenprodukt des Renders ist also eine Karte wie "solid" Objekte wirklich sind und an welchen Stellen sie Transparent sind.



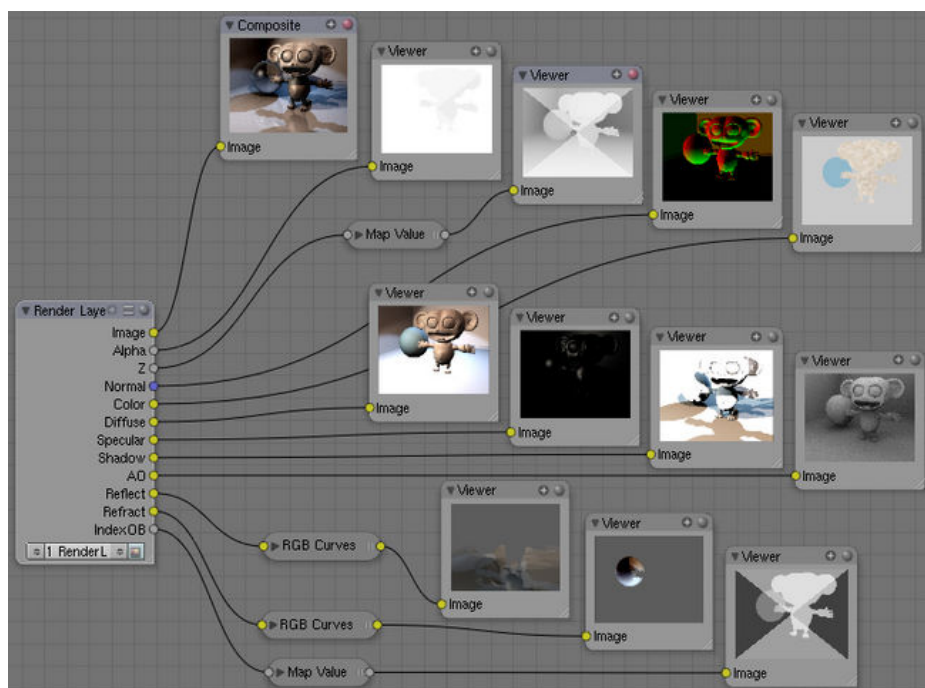
Alpha Werte

Im kleinen Nodenetzwerk oben ist der Alpha Ausgabe Sockel der Renderebenen Node mit einem Kartenwert Node verbunden, welche die Werte in sichtbare Farben umwandelt, und dieser mit einem Anzeige Node, die die Farben dann anzeigt. Vollständig solide Objekte werden weiß und vollständig durchsichtige Teile des Bildes blau dargestellt.

Optionale Sockel

Für die optionalen Sockel müssen die entsprechenden Pässe Eigenschaftenfenster im Renderebenen Panel aktiviert sein.

In einer einfachen Szene zeigen der Suzanne und ein Ball wie welche Pass aussieht:

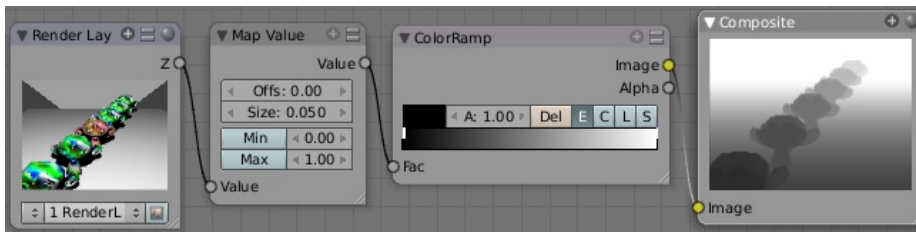


Die verfügbaren Sockel sind und speichern:

- Z Tiefe, Entfernung von der Kamera in Blendereinheiten
- Normalen (Nor) die Normaleninformationen der Flächen
- UV wie die Textur von den UV Koordinaten verzerrt wird
- Geschwindigkeit wie schnell und in welche Richtung das Objekt sich bewegt
- Farbe die RGB Werte
- Diffus welchen Anteil der Diffuse Shader hat
- Specular welchen Anteil der Specular Shader hat
- Schatten die von anderen Objekten geworfen werden
- Umweltverdeckung (Ambient Occlusion) wie die Farben von AO beeinflusst werden.
- Reflektion die Farben die reflektiert werden
- Brechung die Farben die durch die Objekte durchscheinen
- Leuchten Farben die durch Leuchteffekte anderen Objekte entstehen
- Objekt Index den index der Objekte für Auswahl im Nodenetzwerk

Benutzung des Z Wert Sockels

Der Z Wert enthält die Entfernung der Objekte von der Kamera bzw. die Länge des Wegs den der Lichtstrahl zurücklegen musste um zur Kamera zu gelangen. Die Benutzung des Z Werts ist essentiell für realistische Bilder, da Gegenstände die weiter von der Kamera entfernt sind aufgrund der Linse verschwommener sind. Um diese Effekt nachzuahmen müssen die Z Werte verwendet werden. Die Z Werte der Pixel können als Farbkarte dargestellt werden.



Farbkarte der Z Werte

Benutzung des Geschwindigkeitssockels

Der Geschwindigkeitssockel enthält Informationen wie schnell sich Objekte und Vertices in der Szene Bewegen. Obwohl die Objekte in der Szene animiert sind, zeigen einzelne Bilder keinerlei Bewegung. Dafür ist es nötig einen Verschwimmeffekt zu erzeugen wie er bei Bewegungen die

mit einem Film oder vom Auge wahrgenommen werden. Siehe [hier](#).

Bildnode

Panel: [Nodeeditor](#) → [Node Komposition](#)

Menu: ⇧ ShiftA → [Eingabe](#) → Image

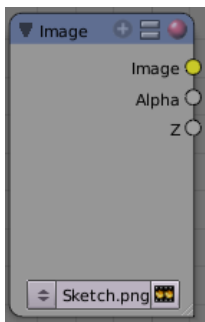


Bild Node

Die Bildnode lädt ein Bild jedes unterstützte Bildformat [format that is supported by Blender](#) oder auch

Alpha, Z Werte und Ebenen falls diese vorhanden sind. Dieser Node kann benutzt werden um:

- Einzelne Bilder (einzel Bilder wie JPG oder einzelne Frames aus Bildsequenzen oder AVI Dateien.
- Das aktuell im UV/Bildeditor geladene Bild

einzufügen.

Bild Kanäle

Ein geladenes Bild hat seine Kanäle als Ausgabesockel auf der rechten Seite. Abhängig vom Format enthalten die folgenden Kanäle sinnvolle

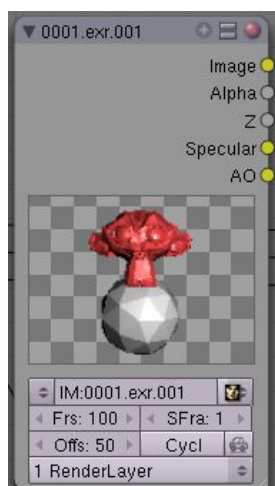
Werte, sonst werden sie auf Standardwerte gesetzt:

- Alphakanal
- Z Wert Kanal

Formate

Blender unterstützt viele Bildformate, allerdings hat von diesen nur das OpenEXR Format RGB Alpha und Z Wert Kanäle.

Speichern und Laden von Renderpässen



Blender kann Render Ebenen und Pässe in einem Multiebenenformat, einer Erweiterung des OpenEXRformats speichern. Dafür müssen alle

gewünschten Renderpässe eingeschaltet sein und alle passenden Einstellungen im Nodeeditor festgelegt sein und als Ausgabeformat das

OpenEXRformat eingestellt sein.

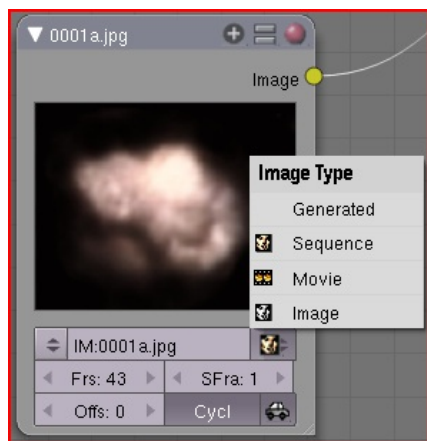
Bildgröße

Größenunterschiede von Bildern können einige Probleme verursachen, deswegen ist es wichtig das alle nötigen Fehler korrekt von den Nodes

ausgeglichen werden.

- Bei unterschiedlichen Bildgrößen in Nodes bestimmt das obere Bild die Ausgabegröße.
- Die Komposition ist standardmäßig zentriert.

Animationen



Bei der Benutzung von Bildsequenzen muss die richtige Eingabeoption benutzt werden. Das Menü gibt folgende Optionen:

1. Erzeugt - eine von Blender erzeugte Bildsequenz
2. Sequenz - lädt eine Sequenz von einzelnen separaten Bildern
3. Film - lädt einen Film in einem .avi oder einem .mov Format
4. Bild - lädt ein normales Bild

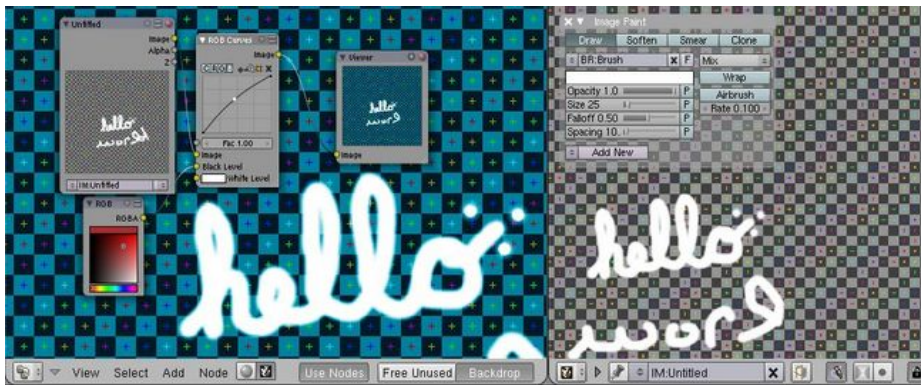
Ein Film oder Bild name kann willkürlich gewählt sein, aber Bildsequenzen müssen eine Nummerierung im Namen haben haben wie

Beispiel0001.jpg, Beispiel0002.jpg, Beispiel0003.jpg um Blender Informationen zur Reihenfolge geben zu können.

Die Verschiedenen Knöpfe

- Frs - stellt die Anzahl der zu verwendenden Frames (Einzelbildern) ein.
- SFra - stellt das Einzelbild ein an dem die Sequenz eingefügt werden soll.
- First - stellt die Nummer des ersten Bildes der Sequenz ein das benutzt werden soll.
- Cycl. - stellt ein das die Sequenz wiederholt werden soll.

Erzeugte Bilder



Benutzung der Nodes zum modifizieren eines Bilds im UV/Bild Fenster

Blenders [Texturbemalen](#) hat bei eingeschaltetem sync lock Auswirkungen über alle Teile von Blender. Das bedeutet das ein bereits geladenes Bild im UV/Bild Editor verändert werden kann und die Auswirkungen direkt im Nodeeditor benutzt werden können.

Sonstiges

Blender korrigiert nicht automatisch nicht passende Einzelbildzahlen von unterschiedlichen Videosequenzen. Eine unterschiedliche Bildfrequenz

muss mit entsprechenden Werkzeugen im [Video Sequenz Editor](#) eingestellt werden.

AVI (Audio Video Interlaced) datei sind kodiert und um sie richtig zu laden ist ein entsprechendes codec nötig. Beim Error

FFMPEG or unsupported video format muss ein entsprechendes Codec installiert werden.

Verbinden von Videosequenzen mit dem Nodeeditor

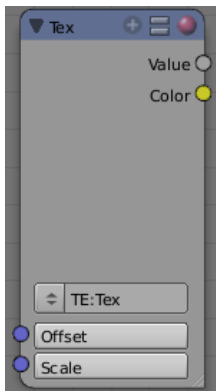
Die oben beschriebenen Kontrollen sind alles was nötig ist um Videosequenzen zu verbinden. Dafür können verschiedene Übergänge im Nodeeditor eingestellt werden:

- ein einfacher Schnitt
- Einblenden
- Ausblenden
- Überblenden
- andere Übergänge

Textur Node

Panel: [Nodeeditor](#) → [Node Komposition](#)

Menu: ⇧ ShiftA → [Eingabe](#) → Textur



Textur Node

Der Textur Node macht dem Nodeeditor 3D Texturen verfügbar. Es können

- Offset (Verschiebung)
- skalierung

der Textur eingestellt werden.

Wert Node

Panel: [Nodeeditor](#) → [Node Komposition](#)

Menu: ⇧ ShiftA → [Eingabe](#) → Wert

Die Wert Node gibt den eingestellten Wert weiter.

RGB Node

Panel: [Nodeeditor](#) → [Node Komposition](#)

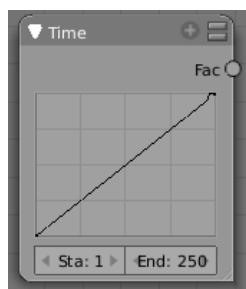
Menu: ⇧ ShiftA → [Eingabe](#) → RGB

Der RGB Node erzeugt eine Farbe zur weiteren Verwendung im Nodeeditor. Dazu können die verschiedenen Methoden zur Farberzeugung benutzt werden: durch Eingabe von Werten (RGB, HSV, und ein Hexcode für die entsprechende Farbe) oder manuelles einstellen der Farbe.

Zeit Node

Panel: [Nodeeditor](#) → [Node Komposition](#)

Menu: ⇧ ShiftA → [Eingabe](#) → Zeit



Zeit Node

Der Zeit Node erzeugt eine Funktion die die ablaufende Zeit auf einen Wert zwischen 0 und 1 abbildet. Die Knöpfe für Start und Ende markieren den Zeitabschnitt der als x Wert benutzt wird. Die Node gibt den y Wert aus.

Das Beispiel zeigt eine Zeitspanne von 250 Frames und eine lineare Funktion. Das bedeutet das der Wert für Bild 0 ist 0 für Bild 125 ist es 0,5 und für Bild 187 ist es 0,75. Die Funktion kann durch bewegen mit der Maus und hinzufügen von Kontrollpunkten verändert werden. Der Wert wird ausserhalb der Zeitspanne linear extrapoliert, deswegen kann es wichtig sein die Werte einzugrenzen. Die Eingabezeitwerte können umgekehrt werden indem ein größerer Startwert für die Einzelbildzahl eingegeben wird als für den Endwert.

Ausgabe Nodes

Es ist hilfreich an beliebigen Stellen im Prozess zu sehen wie ein Zwischenergebnis oder am Ende das Endergebnis aussieht. Dafür werden Ausgabe Nodes benutzt. Sie verbinden alle vorangehenden Nodes zu einer Ausgabe.

Anzeigenodes



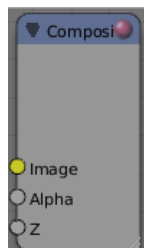
Anzeige Node

Die Anzeigenodes sind Nodes für Zwischenansichten. Sie können für alle Bildausgaben benutzt werden und erzeugen ein Hintergrundbild aus den eingehenden Daten.

Linksklicken macht einen Anzeige Node zum aktiven Node oder aktualisiert die Anzeige.

Es gibt einzelne und Split Anzeige Nodes. Beide erzeugen eine Ausgabe wenn die Box "Hintergrund" aktiviert ist.

Kompositionsnode



Kompositionsnode

Der Kompositionsnode ist der eigentliche Ausgabepunkt des Kompositionsprozesses. Das Bild das hier eingegeben wird ist das fertige Ergebnis.

Es können die Werte für RGB, Alpha und Z hier ausgegeben werden, was nützlich sein kann wenn sie zwischendurch verändert wurden.

Es sollte nur eine einzige Ausgabenode geben. Mehr machen keinen Sinn und können zu unvorhergesehen (und unerwünschten) Ergebnissen führen.

Datei Ausgabe Node



Datei Ausgabe Node

Dieser Node erzeugt Ausgabe Bilder in RGBA im gerenderten Format für alle Bilder der Bildsequenz. Der Dateiname besteht dann aus "Dateiname"+Framezahl+Format.

Abhängig vom ausgewählten Format werden zusätzliche Speicheroptionen verfügbar.

Page status ([reviewing guidelines](#))

Text This should be in the Datamanagement section

Proposed fixes: none

Diese Node kann das Speichern des Renderergebnisses übernehmen, sodass Zwischenergebnisse später verfügbar sind und nicht immer per Hand ausgeführt werden müssen.

Dateidetails

Wie bei anderen Dateinamen verweist // auf das lokale Verzeichnis der .blend Datei.

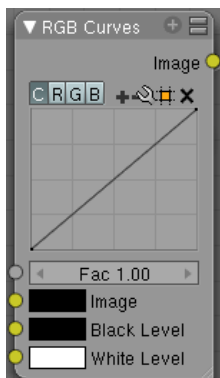
Level Node

Der Level Node nimmt ein Bild und kann Durschnitts-, Standardnormalverteils- und Mittelwerte für RGB und Leucht Kanäle ausgeben.

Farbkompositionsnodes

Diese Nodes verändern Farben oder Farb ähnliche Eigenschaften von Bilder basierend auf den einzelnen Pixeln.

RGB Kurven Node



RGB Kurven Node

Dieser Node verändert den angegebenen Kanal entsprechend der eingestellten Kurve. Wie ein Kurven Node funktioniert kann hier [hier](#) nachgelesen werden.

Mischen Node



Der Mischen Node verändert das Bild im Oberen Sockel mit einem Wert oder Bild im unteren Sockel. Es gibt folgende Modi für den Mischen Node:

Mischen

Der Oberen Wert wird vom unterem mit dem Faktor überdeckt.

Addieren

Addiert die Pixel Werte. Der Faktor steuert den Einfluss.

Subtrahieren

Subtrahiert die Pixel Werte. Der Faktor steuert den Einfluss.

Multiplizieren

Multipliziert beide Farbwerte miteinander.

"Screen"

Beide Pixelwerte werden invertiert, multipliert und das Produkt wieder invertiert.

Überlagerung

Eine Kombination der "Multiplizieren" und "Screen" Methoden, basierend auf der Grundfarbe.

Dividieren

Der Wert aus dem oberen Sockel wird durch den Wert aus dem unteren Sockel geteilt.

Unterschied

Errechnet den absoluten Unterschied der Farbwerte.

Abdunkel

Vergleicht beide Pixelwerte und nimmt den kleineren.

Auffichten

Vergleicht beide Pixelwerte und nimmt den größeren.

"Dodge"

Hellet den ersten Sockel basierend auf dem Gradienten des zweiten Sockels auf.

"Burn"

Verdunkelt den ersten Sockel basierend auf dem Gradienten des zweiten Sockels.

Farbwert

Addiert einen Farbwert.

Helligkeit

Die RGB Werte werden zu HSV Werten umgewandelt, die Helligkeitswerte werden gemittelt und die HSV Werte wieder in RGB umgewandelt.

Sättigung

Die RGB Werte werden zu HSV Werten umgewandelt, die Sättigungswerte werden gemittelt und die HSV Werte wieder in RGB umgewandelt.

Farbton

Die RGB Werte werden zu HSV Werten umgewandelt, die Farbtonwerte werden gemittelt und die HSV Werte wieder in RGB umgewandelt.

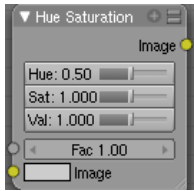
Alpha

Der Alpha Knopf bewirkt das der Alphawert des zweiten Bildes zur Berechnung benutzt wird.

Faktor

Ist der Verrechnungsfaktor der bestimmt wieviel Einfluss eine Operation auf den Farbwert hat.

Farbton Sättigungs Node



Als Alternative zu RGB Bearbeitung kann Farbe auch als Mischung von Farbtönen, Helligkeiten und Sättigungen betrachtet werden. Dies kann

nützlich sein um verblichene Farben hervorzuheben oder einen Verbleichungseffekt zu erzeugen.

Dieser Node verändert einen Eingestellten Farbwert um die eingestellten Werte.

Farbwert:

Der Farbwert bestimmt wieviel die Farben im Bild in welche Richtung des Spektrums verschoben werden.

Sät:

Die Sättigung bestimmt wieviel Farbe in der Farbe enthalten ist.

Helligkeit:

Bestimmt wie stark das Bild aufgehellt oder abgedunkelt wird.

Faktor:

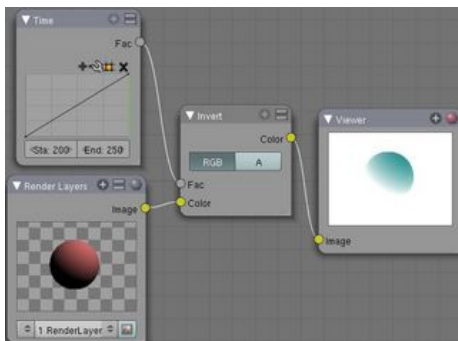
Bestimmt den Gesamteinfluss aller Einstellungen dieser Node.

Gamma



Gamma Korrektur.

Umkehrung



Erzeugt ein Negativ.

Optionen

Faktor

Kontrolliert den Einfluss der Node

Farbe

Eingabefarbe oder Bild

RGB

Invertiert die Farbe mit Weiß als 1 Wert.

A

Invertiert den Alphakanal

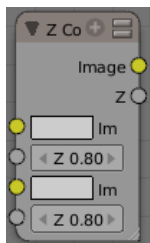
AlphaÜber Node



AlphaÜber
node

Dieser Node kann benutzt werden um zwei Bilder über einander zu legen. Hintergrund ist der obere, Vordergrund der untere Sockel.

Z-Kombinations Node



Z-
Kombinations
Node

Die Z Kombination benutzt den Z Wert von zwei Bildern um zu bestimmen welche Pixel des einen vor Pixeln des anderen Bildes sind. Falls beide

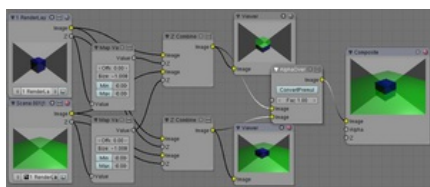
gleich sind, wird der Wert des oberen Sockels genommen.

Alpha values carry over from the input images. Not only is the image pixel chosen, but also its alpha channel value. So, if a pixel is partially or totally transparent, the result of the Z-Combine will also be partially transparent; in which case the background image will show through the foreground (chosen) pixel. Where there are sharp edges or contrast, the alpha map will automatically be anti-aliased to

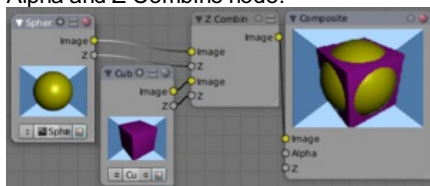
smooth out any artifacts.

However, you can obtain this by making an AlphaOver of two Z-Combine, one normal, the other having inverted (reversed?) Z-values as inputs,

obtained using for each of them a MapValue node with a Size field set to -1.0:



Alpha and Z-Combine node.



Choosing closest pixels



Z-Combine in action

Farbbalance

Dieser Node passt Farbwerte von zwei Bilder aneinander an. Dabei werden zwei unterschiedliche Formeln benutzt:

.Lift, Gamma, Gain Lift hellt dunkle Farben auf, Gamma passt mittlere Farbtöne und Gain

highlights an.

Offset, Power, Slope

Ausgabe ist = $(i*s+o)^p$,

wobei

i = Eingabe Pixelwert (0=schwarz, 1=weiß)
 s = slope (any number 0 or greater, nominal value is 1.0)
 o = offset (any number, nominal value is 0)
 p = power (any number greater than 0, nominal value is 1.0)

Farbkorrektur(HSV)

Dieser Node passt Hue Saturation und Value eines Bildes mittels einer Kurve an.

Farbtonkarte

Diese Node wird in CG und Bildbearbeitung benutzt um ein Farbset auf ein anderes abzubilden, um das Aussehen von Hoch dynamischen Bildern in

weniger dynamischen Reichweiten zu erreichen.

Die Farbtonkarte korrigiert das Problem der starken Kontrastverringern durch Radiancewerte und behält Bilddetails und Farbaussehen bei.

Es gibt zwei Berechnungsmethoden:

Rh Einfach

Schlüssel

Der Wert auf den die durchschnittliche Leuchtkraft abgebildet wird.

Abstand

Normalerweise 1, kann aber dazu benutzt werden die Helligkeitskurve zu verändern

Gamma

Gamma

R/D Photorezeptor

Intensität

Falls der Wert kleiner als 0 ist wird das Bild abgedunkelt, sonst aufgehellt.

Kontrast

Auf 0 wird die Schätzung vom Eingabebild benutzt.

Adaptation

Falls 0, gilt der Wert global; Falls 1, basiert der Wert auf der Pixel Intensität.

Farbkorrektur

Falls 0 gleich für alle Kanäle, bei 1 werden alle unabhängig voneinander korrigiert.

Sequenz-Bearbeitung

Im Zusatz zur Modellierung und Animierung, verfügt Blender über einen voll funktionalen Video-Sequenz-Editor (VSE) als auch über einen erweiterten node-basierten Editor der auch einen Videostrom manipuliert. [Compositing Nodes](#) operate equally well on images or video streams, and can apply detailed image manipulation on the stream.

Operating at a higher conceptual level, and used later in the video production process, Blender's legacy VSE operates on a set of entire strips at a time, as a chunk of footage. Die vielen Teile von Blender arbeiten in dem üblichen Work Flow Style zusammen:

- Model zum Konstruieren der Objekte
- Material Zuweisung und die Einführung von Beleuchtung zur Färbung der Objekte
- Animierung Ihrer Objekte um sie beweglich zu machen
- Das Rendern von Video-Ebenen mittels Kameras
- Die Nutzung von Compositing nodes zum:
 - Verbessern der Bilder durch Adjustierung von Farben, dem hinzufügen von Spezialeffekten in der Szene
 - Schichten der Bilder zu einer Composite-Bild-Sequenz (kürzen)
- Zusammenfügen der Videostreifen um einen Film durch die VSE zusammenzustellen.

Die VSE in Blender ist ein komplettes Videobearbeitungs-System dass es Ihnen erlaubt mehrere Videokanäle zu kombinieren und Effekte zu ihnen hinzuzufügen. Dessen Funktionalität ist schon seit dem Anfang von Blender mit dabei. Auch wenn es eine beschränkte Anzahl an Operationen hat, können Sie diese zum effektiver Videobearbeitung nutzen (besonders wenn sie dass mit der Animierungskraft von Blender kombinieren!) ist es durch ein Plugin System erweiterbar um eine uneingeschränkte Anzahl an Bild Manipulierungen durchzuführen.

Using the VSE, you load multiple video clips and lay them end-to-end (or in some cases, overlay them), inserting fades and transitions to link the end of one clip to the beginning of another. Finally, add an audio track so you can synchronize the timing of the video sequence to match it. The result of using the VSE is your finished movie.

FFMPEG Support

Die Unterstützung für das Exportieren eines avi/quicktime Videos mit Hilfe von FFMPEG funktioniert, momentan (seit 2.44) nur mit den Linux und Windows Versionen. Mit FFMPEG Support, kannst du Audiospuren mit in dein Video abspeichern.

Blender 2.6, Python Handbuch

Einleitung

Willkommen zum Blender 2.6 Python Handbuch.

Über Python

Für Python ist praktisch die ganze Dokumentation auf Englisch oder zumindest auf Englisch eher verfügbar als auf Deutsch. Es gibt zwar deutsche Tutorials und Anleitungen für Python aber die Sprache selbst funktioniert trotzdem auf Englisch. Obwohl es zwar wahrscheinlich möglich ist Python zu beherrschen ohne Englisch zu können, ist es wahrscheinlich sehr viel schwerer. Alle hier verlinkten Seiten, Tutorials und Handbücher sind dementsprechend auf Englisch.

Python www.Python.org ist eine interpretierte interaktive objektorientierte Programmiersprache. Sie enthält Module, Exceptions, dynamisches Tippen, sehr hochlevelige Datentypen und Klassen. Python kombiniert eine beachtliche Funktionalität mit einer sehr gut lesbaren Syntax. Python benutzt englische Schlüsselwörter und Syntax.

Python Skripte sind ein sehr nützlicher und flexibler Weg um Blender Funktionen zu erweitern oder spezieller auszunutzen. Praktisch alles was in Blender durch Menüs oder Funktionen erreichbar ist, ist ebenfalls durch Skripte erreichbar - nur in vielen Fällen viel einfacher oder schneller, besonders aber nicht nur im Fall von einfach beschreibbaren Funktionen. Mit Python sind unter anderem Import Export, Animation, Rendereigenschaften, Objekt und Mesherstellung möglich..

Um mit Blender zu interagieren benutzen Skripte das Anwendungs Programmier Interface (API).

Allgemeine Informationen

Links die beim schreiben von Skripten nützlich sind:

- [Blender Python API](#)
 - Die offizielle API dokumentation.
- [API Einführung](#)
 - Eine kurze Einführung in die API a short introduction to get you started with the API. Contains examples.
- [CookBook](#)
 - A section of handy code snippets (yet to be written)
- [FAQ](#)
 - Frequently asked questions and their answers

Links zum teilen von Skripten:

- [teilen von Skripten](#)
 - Informationen wie man Skripte mit anderen teilt und sie in die offizielle Blender version zu integrieren.
- [Add-Ons erstellen](#)
 - Seit Blender 2.5 Alpha1, gibt es einen neuen Weg Skripte für Blender zu verteilen.
- [Extensions project](#)
 - Projekt zur zentralen Kontrolle über die in Blender enthaltenen Erweiterungen.

Anfang - Wiki Tutorials

Die folgenden Seiten sind auf dieser wiki und führen durch die Grundlagen des Pythonskriptens in Blender:

- [Hello World](#)
- [Console](#)
- [Text editor](#)
- [Geometry](#)
- [Properties, ID-Properties and their differences](#)

Anfang - Externe links

Die folgenden Seiten gehören nicht zu dieser Wiki, enthalten aber eine Menge gute Informationen für das Schreiben von Skripts für Blender. The following pages are not located on this wiki, but contain a lot of good information to start learning how to write scripts for Blender.

- [Introductory tutorial by Satish Goda](#)
 - Ein für Anfänger konzipiertes Tutorial für grundlegende Manipulationen mit der API.
- [Ira Krakow's video tutorials](#)
 - Eine Videotutorialserie
- [Quickstart guide](#)
 - Eine Schnellstartanleitung für erfahrene Benutzer von Blender und Python.
- [Examples thread](#)
 - Ein Forumthread mit vielen funktionierende Skriptbeispielen.
- [Introduction to Python](#)
 - Ein einstündiges Videotutorial das Python und in die Blender API einführt.

Über Python

Da Python eine auf Englisch basierende Programmiersprache ist, ist praktisch die ganze Dokumentation auf Englisch oder zumindest auf Englisch eher verfügbar als auf Deutsch. Es gibt zwar deutsche Tutorials und Anleitungen für Python aber die Sprache selbst funktioniert trotzdem auf Englisch. Obwohl es zwar wahrscheinlich möglich ist Python zu beherrschen ohne Englisch zu können, ist es wahrscheinlich sehr viel schwerer. Alle mit diesem Thema in Verbindung stehen Seiten, Tutorials und Handbücher sind dementsprechend vorerst auf Englisch. Zum Erreichen der englischen Version dieser Seite muss nur oben die Sprache auf Englisch umgestellt werden.

Über Python

Da Python eine auf Englisch basierende Programmiersprache ist, ist praktisch die ganze Dokumentation auf Englisch oder zumindest auf Englisch eher verfügbar als auf Deutsch. Es gibt zwar deutsche Tutorials und Anleitungen für Python aber die Sprache selbst funktioniert trotzdem auf Englisch. Obwohl es zwar wahrscheinlich möglich ist Python zu beherrschen ohne Englisch zu können, ist es wahrscheinlich sehr viel schwerer. Alle mit diesem Thema in Verbindung stehen Seiten, Tutorials und Handbücher sind dementsprechend vorerst auf Englisch. Zum Erreichen der englischen Version dieser Seite muss nur oben die Sprache auf Englisch umgestellt werden.

Über Python

Da Python eine auf Englisch basierende Programmiersprache ist, ist praktisch die ganze Dokumentation auf Englisch oder zumindest auf Englisch eher verfügbar als auf Deutsch. Es gibt zwar deutsche Tutorials und Anleitungen für Python aber die Sprache selbst funktioniert trotzdem auf Englisch. Obwohl es zwar wahrscheinlich möglich ist Python zu beherrschen ohne Englisch zu können, ist es wahrscheinlich sehr viel schwerer. Alle mit diesem Thema in Verbindung stehen Seiten, Tutorials und Handbücher sind dementsprechend vorerst auf Englisch. Zum Erreichen der englischen Version dieser Seite muss nur oben die Sprache auf Englisch umgestellt werden.

Blender Game Engine Überblick

Blender verfügt über eine eigene Game Engine, welche es erlaubt interaktive Anwendungen oder Simulationen in 3D zu erstellen. Der Hauptunterschied zwischen der Game Engine und dem normalen Blender-Modus ist der Render-Vorgang. Im normalen Modus werden die Bilder und Animationen "offline" generiert und können danach nicht mehr verändert werden. Im Gegensatz dazu rendert die Game Engine ihre Szene in Echtzeit und bietet Möglichkeiten, in den Render-Vorgang einzugreifen.

Die Blender Game Engine (BGE) ist damit ein mächtiges "high-level" Programmier-Werkzeug für Spiele entwicklung. Sie kann aber auch dazu genutzt werden um jegliche Art von interaktiver 3D Software für andere Bereiche, wie z.B für interaktive 3D-Architektur-Tools oder die pädagogische Forschung in der Physik, zu erstellen.



Screenshot aus "Yo Frankie", erstellt mit der Blender Game Engine

Eigenschaften der Game Engine

Die BGE enthält eine Hauptschleife, die Logik, Sound, Physiksimulation und Rendering nacheinander bearbeitet. Die Engine ist in C++ geschrieben, bietet aber eine Python-Schnittstelle.

Die BGE ist eng in Blender integriert, was einen schnellen Wechsel zwischen dem traditionellen Modellieren und den Spiel-spezifischen Funktionen ermöglicht. Auf diese Weise kann die Game Engine für alle Stufen der Entwicklung genutzt werden, vom Prototypen bis hin zum fertigen Spiel.

Die Game Engine kann innerhalb von Blender laufen, bietet aber auch die Möglichkeit, eine binäre Datei zum Ausführen unter Windows, Linux und MacOS zu erstellen. Zusätzlich gibt es Ansätze für mobile Geräte mittels des Android Blender Player GSOC 2012-Projektes.

Verschiedene Bibliotheken sind in den 2.5- und 2.6-Releases von Blender enthalten. Darunter:

- Recast: ein State-of-the-Art Konstruktions-Toolset für Navigations-Meshes für Spiele
- Detour: Funktionen für Pfadsuche und räumliche Orientierung
- Bullet: Physik-Engine mit 3D-Kollision, soft body physics und Festkörperdynamik
- Audaspace: Sound Library für OpenAL oder SDL

Nutzung der Game Engine

Der Kern der Struktur der BGE sind die "[Logic Bricks](#)". Das Ziel von Logic Bricks ist das Bereitstellen einer einfach zu benutzenden visuellen Oberfläche zum Entwerfen von interaktiven Anwendungen ohne jegliches Programmierwissen. Der [Logic Editor](#) stellt dafür tief integrierte Interaktion mit der Simulation zur Verfügung. Zusätzlich kann er durch Python-Skripte erweitert werden. Eine Übersicht über den Editor kann im [Game Logic Screen Layout](#) gefunden werden.

Es existieren drei Sorten von Logic Bricks:

- [Sensoren \(Sensors\)](#)
- [Kontroller \(Controllers\)](#)
- [Aktuatoren \(Actuators\)](#)

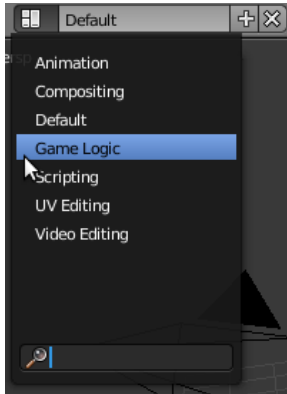
Die Erstellung einer Simulation oder eines Spiels mit BGE umfasst normalerweise vier Schritte:

1. Erstellung von visuellen Elementen, die gerendert werden können. Das können entweder Modelle oder Bilder sein.
2. Einbringen von Interaktion in die Szene mit Logic Bricks, die Verhalten zu definieren und beschreiben, wie es erzeugt wird. Dazu werden "Sensoren" wie Tastatur oder Joystick benutzt.
3. Definition einer (oder mehrerer) Kamera, die einen Blickwinkel für die Szene bereitstellen und Parameter für die Anzeige enthalten.
4. Starten des Spiels, entweder im internen Player oder als exportiertes Binärprogramm für die jeweilige Plattform.

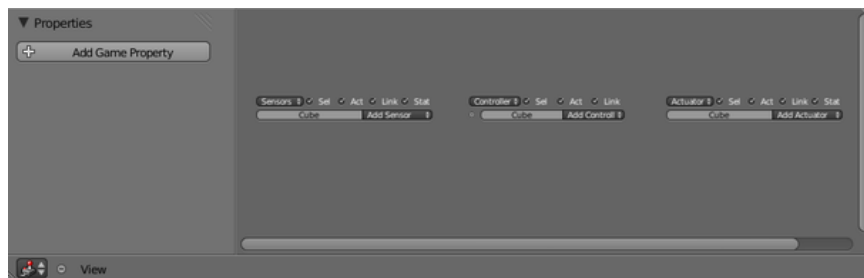
Für Entwickler die Python bevorzugen bietet die BGE ihre eigene [Python API](#), separat vom Rest von Blender, welche dafür genutzt werden kann Skripte zur Kontrolle der Anwendung zu schreiben. Dies wird durch einen [Python Controller](#) und das Linken zu einem Python-Skript realisiert.

Spielelogik

Die Spielelogik ist alles was im Spiel etwas passieren lässt. Sie ist dazu entworfen ein grafisch mächtiges Werkzeug für die Logik bereitzustellen. Die Blöcke (auch "Bricks" genannt) repräsentieren vorprogrammierte Funktionen die verändert und kombiniert werden können, um das Spiel oder die Anwendung zu erstellen. Das System ist in drei Teile aufgeteilt: [Sensoren](#), [Kontroller](#) und [Aktuatoren](#). Sensoren erkennen Veränderungen, wie Kollisionen, einen Tastendruck, einen Mausklick. Sensoren sind [verbunden](#) mit Kontrollern, welche vergleichen und Aktuatoren aktivieren.

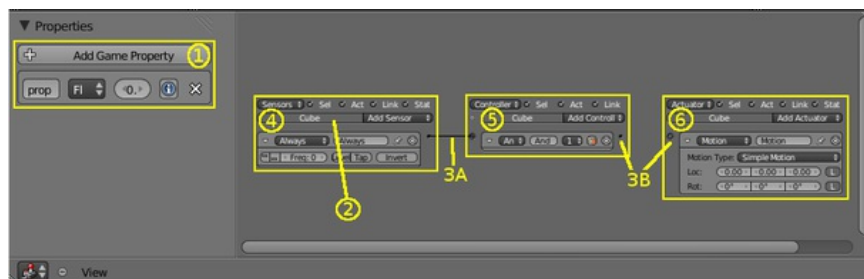


Sie können das Logik Panels (kontext) für das erste Mal, entweder durch Drücken von **Ctrl→** oder durch Auswählen von Game Logic aus dem Layout Menü, laden.



Das Logikpanel.

Um Ihnen ein besseres Verständnis vom Logikpanel zu geben, wurden einige Menüs erweitert und in dem unterem Bild durchnummeriert. Wir werden uns jeden Abschnitt individuell im unterem Bild ansehen.



Die verschiedenen Teile des Logikpanels.

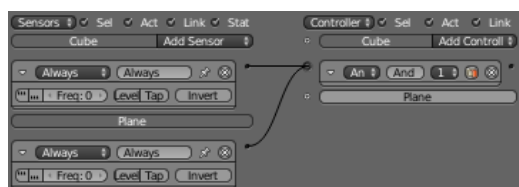
Eigenschaften

Eigenschaften sind wie Variablen in anderen Programmiersprachen. Sie werden verwendet, um mit einem Objekt zugeordnete Daten zu speichern und auf sie zuzugreifen. Die folgenden Typen sind verfügbar:

- Timer - Startet ab einer definierten Zahl und zählt hoch.
- String - Speichert Text, Zeichenketten
- Float - Speichert Dezimalzahlen zwischen -10000,000 und 10000,000.
- Integer - Speichert Ganzzahlen zwischen -10000 und 10000.
- Bool - Ist entweder "TRUE", für wahr, oder "FALSE", für falsch.

Für einen tieferen Einblick, siehe [Eigenschaften](#).


Zugeordnete Objekte



Logik für verschiedene Objekte.

Die Logik in der Blender Game Engine wird an Objekte angehängt. Objekte werden nach ihrem Namen aufgelistet und erscheinen in dem Logikfenster, wenn sie ausgewählt sind. Sie können ein einzelnes Objekt wie oben abgebildet auswählen, oder Sie können mehrere Objekte wie rechts abgebildet auswählen. Die Optionen Sel, Act, link zeigen ausgewählte Objekte, aktive Objekte und verknüpfte Logik in der Reihe an. Die Logik eines Objektes ist nur dann sichtbar wenn das Objekt mit der zugeordneten Logik ausgewählt ist.

Verbindungen

Verbindungen (3A) sind die Richtung des logischen Flusses zwischen Objekten. Verbindungslinien werden durch LMB  gezogen, der von einem Verbindungsknoten (3B) zu einem anderen gezogen wird. Verbindungen können vom Sensoren bis zu Kontrollern, von Controllern zu Aktuatoren oder direkt zwischen Sensoren und Controller gezogen werden. Die letzte Auswahl fügt einen AND Controller als ein Verbinderblock hinzu. Sender-Knoten (die schwarzen Kreise auf der rechten Seite der Sensoren und Controller) können an mehrere Empfänger-Knoten (auf der linken Seite der Controller und Aktuatoren) senden. Empfänger-Knoten können ebenfalls mehrfache Verbindungen erhalten.

Verbindungen können durch halten von Ctrl + LMB  und auswählen der Verbindung gelöscht werden (ähnlich einem Messerschnitt)

Sie können Sensoren mit Aktuatoren nicht direkt verbinden - Ein Controller muss vorhanden sein (Aber wie erwähnt, muss zwischen diesen ein Controller hinzugefügt und verbunden werden). Aktuatoren können nicht zu einem Sensoren zurück verbunden werden; wenn Sie wünschen, dass ein Sensor, nachdem ein Aktor beendet wurde, aktiviert wird, verwenden Sie den Aktor-Sensor.

Sensoren

Sensoren starten alle Logikhandlungen. Sensoren können Dinge sein, wie ein nahe gelegenes Objekt, eine gedrückte Taste auf der Tastatur, zeitlich festgelegte Ereignisse usw.. Wenn ein Sensor ausgelöst wird, wird ein Impuls an alle verbundenen Controller gesendet.

Für einen tieferen Einblick, siehe [Sensoren](#).

Kontroller

Kontroller handhaben die Logik, werten Impulse von Sensoren aus und senden als Reaktion darauf Impulse an Aktuatoren weiter. Die verschiedenen Arten von Controllern sind:

- AND - Alle verbunden Sensoren müssen positiv sein um einen positiven Impuls zu senden.
- OR - Einer oder mehrere Sensoren müssen positiv sein.
- XOR - Exklusives Or: einer, und nur einer, der verbundener Sensoren muss positiv sein.
- NAND - Not And, umgedrehter AND Controller.
- NOR - Not Or, umgedrehter OR Controller.
- XNOR - Exklusiver NOR Controller.
- Ausdruck - Schreiben Sie Ihre eigene Erweiterung.
- Python - Kontrollieren Sie den Sensor mit einem Python-Skript oder Modul.

Für einen tieferen Einblick, siehe [Kontroller](#).

Aktuatoren

Aktuatoren beeinflussen Objekte oder das Spiel in gewisser Weise. Aktuatoren ändern Bewegungen, Sounds, Eigenschaften, Objekte, usw. . Diese sind aus dem allgemeinen Sprachgebrauch mit Auslösern zu vergleichen, führen hier aber etwas aus. Diese Änderungen können in anderen Objekten, anderer Physik, anderen Eigenschaften vorhanden sein, oder sie können Trigger-Ereignisse für andere Logik Bricks verursachen.

Für einen tieferen Einblick, siehe [Aktor](#).

Eigenschaften

Eigenschaften sind die Spielelogik zu Variablen. Sie werden mit dem Objekt gespeichert, und können genutzt werden um Dinge zu repräsentieren wie Munition, Gesundheit, Name, und so weiter.

Typen von Eigenschaften

Es gibt fünf Typen von Eigenschaften:

Timer	Starts at the property value and counts upwards as long as the object exists. It can for example be used if you want to know how long time it takes the player to complete a level.	
Float	Nutzt dezimale Nummern als Werte, mit einer Reichweite von -10000.000 bis 10000.000. Er ist nützlich für präzise Werte.	
Integer	Nutzt Integer-Werte, Ganzzahlen als Werte, mit einer Reichweite von -10000 bis 10000. Dieser Typ ist nützlich für die Aufzählung von Dingen wie Munition, wo Dezimalzahlen unnötig sind.	String Takes text as value. Can store 128 characters.
Boolean	Boolean variable, has two values: true or false. This is useful for things that have only two modes, like a light switch.	

Using Properties

Properties can be set up and initialised in the Properties panel of the Logic Editor - see the [Property Editing](#) page for details. When a game is running, values of properties are set, manipulated, and evaluated using the [Property Sensor](#) and the [Property Actuator](#).

Sensoren

Sensoren sind die logischen Bauteile, **Logik Bricks**, die die Logik dazu veranlassen etwas zu tun. Sensoren liefern eine Ausgabe wenn etwas passiert ist, z.B ein ausgelöstes Ereignis wie die Kollision zweier Objekte, einen Tastendruck auf der Tastatur, oder einen Timer für ein zeitlich abgelaufenes Ereignis. Wenn ein Sensor ausgelöst wird, wird ein positiver Impuls an alle Controller geschickt die mit ihm verbunden sind.

Die logischen Blöcke für alle Typen von Sensoren können durch den [Logik Editor](#) konstruiert und verändert werden; Details über diesen Vorgang finden sich auf der Seite [Sensor Bearbeitung \(Achtung, befindet sich auf einer Benutzerseite\)](#).

Die folgenden Typen an Sensoren sind momentan verfügbar:

Aktuator	Erkennt wenn ein bestimmter Aktuator einen Aktivierungsimpuls erhält.
Immer	Gibt ein fortlaufendes Ausgangssignal aus bei regulären Intervallen.
Kollision	Erkennt Kollisionen zwischen Objekten oder Materialien.
Verzögerung	Verzögert die Ausgabe nach einer festgelegten Nummer an logischen Ticks.
Joystick	Erkennt die Bewegung von bestimmten Joystick Controllern.
Tastatur	Erkennt Tastatureingaben.
Nachricht	Erkennt entweder Textnachrichten oder Eigenschaftswerte.
Maus	Erkennt Maus-Ereignisse.
Nähe	Erkennt Objekte die sich mit einem spezifischen Abstand von sich selber wegbewegen.
Eigenschaft	Erkennt Veränderungen in den Eigenschaften seines Elternobjektes.
Radar	Erkennt Objekte die sich mit einem spezifischen Abstand von sich selber wegbewegen, innerhalb eines Winkels einer Achse.
Zufall	Generiert zufällige Impulse.
Strahl	Schießt einen Strahl in Richtung einer Achse und erkennt Treffer.
Berührung	Erkennt ob das Objekt in Kontakt mit einem anderen Objekt ist.

Aktuator Sensor



Aktuator Sensor

Der Aktuator Sensor reagiert auf das ausführen eines bestimmten Aktuators.

Der Aktuator Sensor sendet einen TRUE Logik-Puls wenn der ausgewählte Aktuator aktiviert wird.

Der Sensor sendet einen FALSE Logik-Puls wenn der ausgewählte Aktuator nicht aktiviert ist.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

Besondere Optionen:

Aktuator

Name des Aktuators (der Aktuator muss zum selben Objekt gehören wie der Sensor.)

Immer Sensor



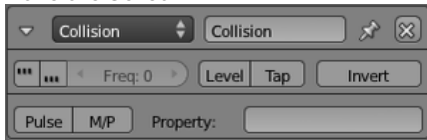
Always sensor

Der Immer Sensor sendet immer einen positiven logischen Puls aus, abhängig von seinen sonstigen allgemeinen Einstellungen. Er wird für alle Dinge verwendet die in jedem Game Engine Tick passieren sollen, oder in regelmäßigen Abständen. Beispiele dafür wären Skript Ausführung, die in jedem Tick passieren sollten, und eine Spielinterne Uhr die nur jede volle Sekunde ihren Wert ändern muss.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

Dieser Sensor hat keine besonderen Eigenschaften oder Einstellungen.

Kollisions Sensor



Collision sensor

Der Kollisions Sensor funktioniert wie ein Touch Sensor kann aber zusätzlich nach Eigenschaften oder Material filtern. Nur Objekte die eine Eigenschaft oder ein Material des angegebenen Names haben lösen diesen Sensor aus. Ein frei gelassenes Namesfeld erzeugt für alle kollidieren Objekte einen positiven Puls.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

Spezielle Optionen:

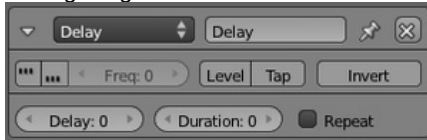
Pulse button

Macht den Sensor scharf für andere mögliche Kollisionen selbst wenn das Sensor Objekt immer noch in Berührung mit dem Objekt ist das den letzten Puls verursacht hat.

M/P button

Wechselt ob Material oder Eigenschaft mit dem Angegebenen Namen gefiltert werden soll.

Verzögerungs Sensor



Delay sensor

Der Verzögerungs Sensor verzögert seinen positive Puls um die angegebene Zeitlänge in Ticks. Das ist nützlich für Prozesse die hintereinander ausgeführt werden müssen.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

Spezielle Optionen:

Delay

Die Anzahl an Ticks um die der Sensor einen positiven Puls verzögert

Duration

Die Anzahl an Ticks um die der Sensor einen negativen Puls verzögert.

Repeat Button

Bewirkt das sich der Sensor zurück setzt und erneut sendet nachdem die Verzögerung vorbei ist.

Joystick Sensor



Joystick sensor

Der Joystick Sensor sendet wenn der Joystick bewegt wird oder ein Button am Joystick gedrückt wird. Es können mehr als ein Joystick benutzt werden (siehe "Index"). Das Verhalten des Joysticks hängt vom Model ab.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

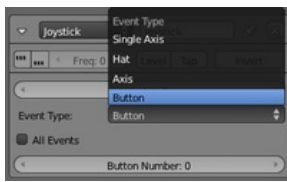
Spezielle Optionen:

Index

Spezifiziert welcher Joystick hier gemeint ist.

All Events

Der Sensor löst aus für jedes mögliche Ereignis des Joysticks.



Joystick Events

Event Type

Spezifiziert welches Joystick Ereignis Auslöser sein soll.



Joystick Single Axis

Single Axis

Erkennt Bewegungen des Joysticks auf einer Achse des Joysticks.

Achsen Nummern

- 1 = Horizontale (links/rechts)
- 2 = Vertikale (vor/zurück)
- 3 = Paddle (hoch/runter?)
- 4 = Verdrehung des Joysticks (gegen den Uhrzeigersinn/im Uhrzeigersinn)

Axis Threshold

Grenzwert bei dem der Joystick ausgelöst wird (?)(fires) (Werte im Bereich von 0 - 32768)



Joystick Hat

Hat

Erkennt Bewegungen eines Knaufs am Joystick.

Hat Number

Spezifiziert welcher Knauf gemeint ist (max. 2)

Hat Direction

Spezifiziert die Richtung in die der Knauf bewegt werden muss um ein Signal auszulösen: oben, unten, links, rechts, oben links, oben rechts, unten links, unten rechts.



Joystick Axis

Axis

Axis Number

Spezifiziert die Achse(1 oder 2)

Axis Threshold

Grenzwert bei dem der Joystick ausgelöst wird (?)(fires) (Werte im Bereich von 0 - 32768)

Axis Direction

Sepzifiziert welche Richtung Auslöser sein soll:

(Achse Nummer = 1) Joystick links, rechts, oben, unten

(Achse Nummer = 2) Paddle oben (links); paddle unten (rechts); Joystick verdrehung links (oben) Joystick verdrehung rechts (unten)

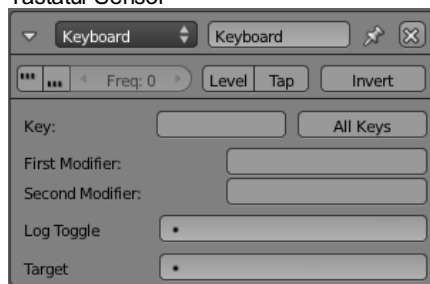


Joystick Button

Button

Spezifiziert die Button nummer die Auslöser sein soll.

Tastatur Sensor



Keyboard sensor

Der Tastatur Sensor erkennt Tastatureingaben und kann diese unter anderem als [String property](#) speichern.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren

Spezielle Optionen:

Key

Dieses Feld enthält den Namen der Taste die den TRUE Puls auslösen soll, wenn sie gedrückt wird. Darauf klicken und eine Taste drücken weist die Taste dem Sensor zu. Auf das Feld klicken und dann daneben löst diese Zuweisung.

Ein FALSE Puls wird gesendet wenn die Taste losgelassen wird.

All keys button

Sendet einen TRUE Puls wenn irgendeine Taste gedrückt wird. Das ist nützlich für Skript gesteuerte Tastenbelegungen [Python controller](#).

First Modifier

Second Modifier

Spezifiziert zusätzlich Tasten die gedrückt sein müssen damit der Sensor einen TRUE Puls aussendet. So können Tastenkombinationen als Auslöser eingestellt werden wie zum Beispiel CtrlR oder {{Shortcut|shift|alt|Esc}}. Sie werden auf die gleiche Weise ausgewählt wie die erste Taste.

LogToggle

Weist einen Bool der als Ein- bzw. Ausschalter für das Aufzeichnen von Tastendrücken in der angegebenen String Eigenschaft dient.

Target

Name der Eigenschaft in der gedrückte Tasten gespeichert werden. Es muss eine Eigenschaft vom Typ String sein. Zusammen mit einem Property Sensor kann das zum Überprüfen von Passwörtern genutzt werden.

Nachricht Sensor



Nachricht Sensor

Der Nachricht Sensor kann Nachrichten oder Eigenschaftenwerte erkennen. Der Sensor sendet einen positiven Puls wenn eine entsprechende Nachricht oder Eigenschaft entdeckt wird. Es kann ein spezielles Thema eingestellt werden auf das der Sensor reagieren soll.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

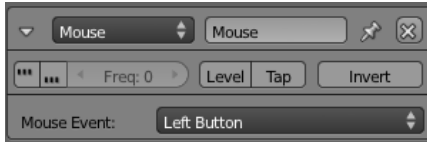
Spezielle Optionen

Subject

Spezifiziert die Nachricht der erhalten werden muss um einen Puls auszusenden. Diese Feld kann ausgelassen werden.

Siehe [Message Actuator](#) für Informationen wie man eine Nachricht verschickt.

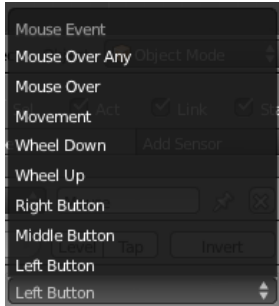
Maus Sensor



Mouse sensor

Der Maus Sensor erkennt Maus-Ereignisse.

Siehe [Sensor Common Options](#) für allgemeine Optionen.



Maus-Ereignisse

Spezielle Optionen

Der Controller besteht nur aus einer Liste von Typen an Maus-Ereignissen. Diese sind:

- Mouse over any, gives a TRUE pulse if the mouse moves over any game object.
- Mouse over, gives a TRUE pulse if the mouse moves over the owner object.
- Movement, any movement with the mouse causes a stream of TRUE pulses.
- Wheel Down, causes a stream of TRUE pulses as the scroll wheel of the mouse moves down.
- Wheel Up, causes a stream of TRUE pulses as the scroll wheel of the mouse moves up.
- Right button gives a TRUE pulse.
- Middle button gives a TRUE pulse.
- Left button gives a TRUE pulse.

Ein FALSE (UNWAHR) Impuls wird geliefert wenn irgendeine der obigen Bedingungen endet.

Es existiert kein Logic Brick für spezifische Mausbewegungen und Reaktionen (so wie die Kamera aus First Person Sicht), diese müssen mit Python realisiert werden.

Nähe Sensor



Nähe Sensor

Ein Nähe Sensor erkennt Objekte welche in einer bestimmten Entfernung vom Objekt, zu dem der Sensor gehört, sind. Es kann wie der Kollisions Sensor nach bestimmten Eigenschaften filtern.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

Spezielle Optionen

Property

Hier kann festgelegt werden das der Sensor nur auf Objekte mit der angegebenen Eigenschaft reagiert.

Distance

Die Anzahl an Blender Einheiten innerhalb der Sensor nach Objekten sucht.

Reset

Die Entfernung die Objekte vom Sensor Objekt entfernt sein müssen um den Sensor zurück zu setzen und einen FALSE Puls auszulösen.

Sonstiges

- 1) Der Nähe Sensor kann durch Objekte nach denen er nicht sucht durch "sehen" wie zum Beispiel Wände.
- 2) Objekte nach denen gesucht wird müssen als Akteur ("Actor") in den Physikeinstellungen eingestellt sein um entdeckt zu werden.

Eigenschaft Sensor

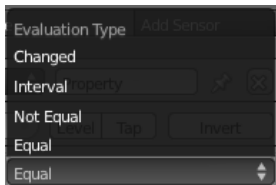


Eigenschaften sensor

Der Eigenschaft Sensor erkennt veränderungen in den Eigenschaften des Objekts zu dem er gehört.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

Spezielle Optionen



Eigenschaften Überprüfung

Evaluation Type

Speziiert auf welche Weise die Eigenschaft überprüft wird.

Changed

Sendet eine positiven Puls aus wenn der Wert sich verändert.

Interval

Sendet einen positiven Puls aus wenn der Wert zwischen Min und Max Werten des Sensors liegt.

Not Equal

Sendet einen positiven Puls aus wenn die Eigenschaft **nicht gleich** dem angegebenen Wert ist.

Equal

Sendet einen positiven Puls aus wenn die Eigenschaft **gleich** dem angegebenen Wert ist.

Um andere Eigenschaften Werte mit dem ersten angegebenen zu vergleichen muss anstelle eines Werts der Name der 2. Eigenschaft angegeben werden.

Radar Sensor



Radar Sensor

Der Radar Sensor funktioniert ähnlich wie ein Nähe Sensor ist aber begrenzt auf einen Kegel, um eine Achse mit einem Öffnungswinkel, mit der Spitze am Radar Objekt und der Basis in der angegebenen Entfernung vom Objekt.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

Spezielle Optionen

Property

Hier kann festgelegt werden das der Sensor nur auf Objekte mit der angegebenen Eigenschaft reagiert.

Sonstiges

- 1) Der Nähe Sensor kann durch Objekte nach denen er nicht sucht durch "sehen" wie zum Beispiel Wände.
- 2) Objekte nach denen gesucht wird müssen als Akteur ("Actor") in den Physikeinstellungen eingestellt sein um entdeckt zu werden.

Axis

Legt die Höhe des Kegels auf eine angegebene lokale Achse des Objekts. Das \pm Zeichen gibt die Richtung an.

Angle

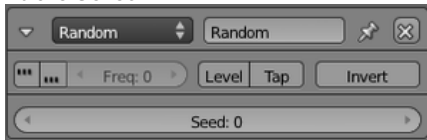
Legt den Winkel im Bereich von 0.00° bis 179.9° des Kegels fest.

Distance

Legt die Höhe des Kegels bzw. Reichweite des Sensors in Blender Einheiten fest.

Dieser Sensor ist zum Beispiel gut um Bots nur ein begrenztes Sichtfeld zu geben.

Zufalls Sensor



Zufalls Sensor

Der Zufalls Sensor erzeugt zufällige Pulse.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

Spezielle Optionen:

Seed

Dieses Feld wird benutzt um den Anfang (Seed) des Zufallsgenerators zu setzen (Werte im Bereich von 0-1000).

Bemerkungen -

- 1) 0 ist nicht Zufällig aber nützlich für Test- und Debugzwecke.
- 2) Wenn der gleiche Seed mehrfach benutzt werden, werden die selben Zahlen herauskommen, allerdings mit zufälligen Intervallen.

Strahl Sensor



Strahl sensor

Der Strahl Sensor sucht nach dem ersten Objekt auf einer Linie von Objekt aus. Es kann nur nach Objekten mit bestimmten Eigenschaften gesucht werden, dann werden alle Objekte die diese Eigenschaft nicht haben nicht beachtet und es wird weitergesucht.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

Spezielle Optionen: Der Strahl Sensor hat viel mit dem [Radar](#) Sensor gemein.

Property

Hier kann festgelegt werden das der Sensor nur auf Objekte mit der angegebenen Eigenschaft reagiert.

Sonstiges

- 1) Der Nähe Sensor kann durch Objekte nach denen er nicht sucht durch "sehen" wie zum Beispiel Wände.
- 2) Objekte nach denen gesucht wird müssen als Akteur ("Actor") in den Physikeinstellungen eingestellt sein um entdeckt zu werden.

Axis

Legt die Höhe des Kegels auf eine angegebene lokale Achse des Objekts. Das \pm Zeichen gibt die Richtung an.

Range

Legt die Reichweite in der gesucht wird in Blender Einheiten fest.

X-Ray Mode button

Stellt den Sensor so ein das er nur auf Objekte mit der eingestellten Eigenschaft oder dem Material reagiert und durch alle anderen hindurchsieht.

Berührungs Sensor



Berührungs Sensor

Der Berührungs Sensor sendet einen positiven Puls wenn das Objekt zu dem er gehört ein anderes Objekt berührt.

Siehe [Allgemeine Sensor Optionen](#) für mehr Informationen zu Sensoren.

Spezielle Optionen

Material

Gibt das Material an, wenn nur Berührungen mit einem Bestimmten Material einen Puls auslösen soll. Wird das Feld freigelassen löst eine Berührung mit einem beliebigen Objekt einen Puls aus.

Ein TRUE Puls wird gesendet wenn die Objekte sich berühren und ein FALSE Puls wird gesendet wenn die Berührung vorbei ist.

Controller

Controller sind Bausteine, welche die Signale von Sensoren verarbeiten und diese an die "Auslöser" ([Actuator](#)) weiterleiten. Sie dienen daher der Verarbeitung und sind "State"-Abhängig. Nach Ausführung der jeweiligen logischen Operation senden sie ein Puls-Signal aus um den "Auslöser", mit welchem sie verbunden sind, anzusteuern.

Wird ein Sensor aktiviert, sendet dieser ein positiven Puls, und wenn er deaktiviert wird, sendet er einen negativen Puls. Die Aufgabe des Controllers ist es die Signale zu kombinieren und zu überprüfen, um sie zum gewünschten Ausgang zu bringen.

Die Logikbausteine und all ihre Typen können im [Logic Editor \(Englisch\)](#) erstellt, gelöscht und geändert werden. Details für diesen Prozess sind hier: [Controller Editing \(Englisch\)](#) zu finden.

Controller Typen

Dies sind die acht Controller-Typen, welche auf separaten, unten stehenden Seiten behandelt werden:

- [AND - UND](#)
- [OR - ODER](#)
- [XOR - UND ODER](#)
- [NAND - NICHT UND](#)
- [NOR - NICHT ODER](#)
- [XNOR - NICHT UND ODER](#)
- [Logischer Ausdruck](#)
- [Python \(Englisch\)](#)

Dies ist eine Tabelle, welche einen kurzen Überblick über die logischen Operationen gibt, die von den entsprechenden Controllern ausgeführt werden können. Die erste Spalte, der Eingang, repräsentiert die Anzahl der positiven Eingangssignale, die von den verbundenen Sensoren gesendet werden. Die folgenden Spalten repräsentieren, wie der Controller auf die Signale reagiert und ausgibt. "Wahr" besagt, dass alle Bedingungen erfüllt sind und der/die angeschlossene(n) "Auslöser" aktiviert wird. "Falsch" besagt, dass die Bedingungen nicht erfüllt sind. Es wird nichts passieren und keine Aktion wird ausgeführt. Bitte besuchen Sie die individuelle Seite der Controller um mehr über diese zu erfahren.

Note

Es wird angenommen, dass mehr als ein Sensor mit dem Controller verbunden ist. Mit Ausnahme der Zeile "Alle".

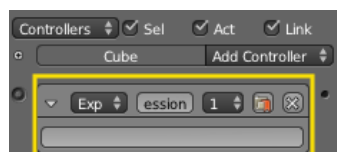
Positive Sensoren	Controller					
	UND	ODER	UND ODER	NICHT UND	NICHT ODER	NICHT UND ODER
Keiner	Falsch	Falsch	Falsch	Wahr	Wahr	Wahr
Einer	Falsch	Wahr	Wahr	Wahr	Falsch	Falsch
Mehrere, nicht alle	Falsch	Wahr	Falsch	Wahr	Falsch	Wahr
Alle	Wahr	Wahr	Falsch	Falsch	Falsch	Wahr

Ausdruck Kontroller

Dieser Kontroller wertet einen vom Benutzer geschriebenen Ausdruck aus, und liefert eine positive (WAHR) Ausgabe falls

Das Ergebnis des Ausdrucks WAHR ist, und
Das Objekt sich im benannten Zustand befindet.

Für alle anderen Bedingungen liefert der Kontroller eine negative (FALSCHE) Ausgabe.



Expression Controller

Ausdruck

Der Ausdruck, der sich in der Box befindet, kann aus Variablen, Konstanten und Operatoren bestehen. Diese müssen den folgenden Regeln folgen.

Variablen

Sie können folgendes nutzen:

- **Sensornamen**,
- **Eigenschaften**: zuweisen einer Spieleigenschaft an ein Objekt und Benutzung in einem Kontroller Ausdruck.

Diese dürfen keine Leerzeichen enthalten.

Operationen

Mathematische Operationen

Operatoren: *, /, +, -

Liefert: eine Nummer

Beispiele: 3 + 2, 35 / 5

Logische Operationen

- Vergleichs Operatoren: <, >, >=, <=, ==, !=
- Boolesche Operatoren: AND (UND), OR (ODER), NOT (NICHT)

Liefert: Wahr oder Falsch.

Beispiele: 3 > 2 (Wahr), 1 AND 0 (Falsch)

Bedingungen (if)

Benutzen Sie:

```
if( ausdruck, impuls_falls_ausdruck_wahr_ist, impuls_falls_ausdruck_falsch_ist )
```

Wenn der Kontroller **ausdruck** als Wahr auswertet:

- wenn **impuls_falls_ausdruck_wahr_ist** gleich **Wahr** ist, sendet der Kontroller einen positiven Impuls an die verbundenen Aktuatoren.
- wenn **impuls_falls_ausdruck_wahr_ist** gleich **Falsch** ist, sendet der Kontroller einen negativen Impuls an die verbundenen Aktuatoren.

Wenn der Kontroller **ausdruck** als Falsch auswertet:

- wenn **impuls_falls_ausdruck_falsch_ist** gleich **Wahr** ist, sendet der Kontroller einen positiven Impuls an die verbundenen Aktuatoren.
- wenn **impuls_falls_ausdruck_falsch_ist** gleich **Falsch** ist, sendet der Kontroller einen negativen Impuls an die verbundenen Aktuatoren.

Beispiele

Ausgehend davon, dass das Objekt über die Eigenschaft **münzen** gleich 30 verfügt:

```
münzen > 20
```

liefert Wahr (der Kontroller sendet einen positiven Impuls zu den verbundenen Aktuatoren).

Ausgehend davon, dass das Objekt über folgendes verfügt:

- einen Sensor namens **schlüssel_Eingesteckt** gleich Wahr,
- eine Eigenschaft namens **Kraftstoff** gleich Falsch,

Schlüssel_Eingesteckt AND Kraftstoff

liefert Falsch (der Controller sendet einen negativen Impuls zu den verbundenen Aktuatoren).

Dass ist das Selbe wie:

```
if (Schlüssel_Eingesteckt AND Kraftstoff, True, False)
```

Anstelle, könnten Sie folgendes tun:

```
if (Schlüssel_Eingesteckt AND Kraftstoff, False, True)
```

um einen positiven Impuls zurückzuliefern wenn **schlüssel_Eingesteckt** **AND** **Kraftstoff** Falsch ausgibt.

Sie können auch folgendes tun:

```
if ((Schlüssel_Eingesteckt AND Kraftstoff) OR (münzen > 20), True, False)
```

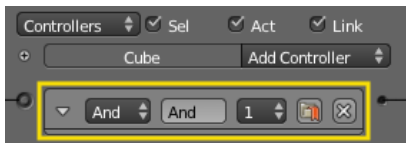
Dieser Ausdruck liefert Wahr, daher sendet der Controller in diesem Fall einen positiven Impuls an alle verbundenen Aktuatoren.

AND/ UND Controller

Dieser Controller erzeugt einen positiven (TRUE/Wahr) Ausgang, wenn:

- Alle Eingänge sind TRUE und
- Alle Objekte sind im gleichen "State"(Ebene).
- Für alle anderen Bedingungen erzeugt der Controller einen negativen (FALSE) Ausgang.

Optionen:



AND Controller

Controller Type menu

Gibt den Typ des Controllers an/ stellt diesen ein.

Controller Name

Der Name des Controllers, welcher vom Nutzer individuell eingestellt werden kann. Der Name wird zur Identifikation mit Python genutzt und muss für das gewählte Objekt einzigartig sein.

State Index

Setzt den vorgesehenen "State"(Ebene), in welchem der Controller operieren wird.

Preference Button

Ist dieser markiert, wird der Controller vor allen nicht markierten Controllern ausgeführt werden (nützlich für start-up-scripts).



Button

Löscht den Controller.

Auslöser

Auslöser führen Aktionen aus, wie Bewegungen, das Erschaffen von Objekten, oder das Abspielen von Klängen. Die Aktuatoren initialisieren ihre Funktionen wenn sie einen positiven Impuls von einem (oder mehreren) Kontroller(n) bekommen.

Die logischen Blöcke für alle Typen von Auslösern können durch den [Logik Editor](#) konstruiert und verändert werden; Details dazu liefert die Seite [Auslöser Bearbeitung](#).

Die folgenden Typen von Auslösern sind aktuell verfügbar:

Aktion	Verwaltet Aktionen des Akteurs, dies ist nur sichtbar wenn ein Akteur ausgewählt ist.
Kamera	Enthält Optionen zum geschmeidigem Folgen des Objekts, primär für Kamera Objekte aber jegliche Objekte können dies benutzen.
Beschränkung	Beschränkungen werden dazu genutzt die Position, Distanz oder Rotation von Objekten einzugrenzen. Diese sind nützlich zum Kontrollieren der Physik von Objekten im Spiel.
Objekt bearbeiten	Bearbeiten des Objekt Mesh, hinzufügen von Objekten oder deren Zerstörung, es kann auch das Mesh des Objektes ändern (und bald auch das Kollisions Mesh neu erstellen).
Filter 2D	Filter für Spezialeffekte wie Sepia Farben oder Verwischung (Blur).
Spiel	Verwaltet das ganze Spiel und führt Dinge wie Neustarts, Beenden, Laden und Speichern durch.
Nachricht	Sendet Nachrichten, welche von anderen Objekten erhalten werden können um sie zu aktivieren.
Bewegung	Versetzt ein Objekt in Bewegung und/oder Rotation, es existieren verschiedene Optionen vom "Teleportieren" zum physikalischen Bewegen von rotierenden Objekten.
Elternteil	Kann dem Objekt ein Elternteil hinzufügen, oder wegnehmen.
Eigenschaft	Manipuliert die Eigenschaften von Objekten, wie Zuweisen, Hinzufügen oder Kopieren..
Zufällig	Erstellt zufällige Werte die in Eigenschaften gespeichert werden können.
Szene	Verwalten die Szenen in Ihrer .blend Datei, diese können als Level oder Benutzeroberflächen und Hintergrundbilder genutzt werden.
Klang	Wird zum Abspielen von Klängen im Spiel genutzt.
Zustand	Ändert Zustände des Objektes.
Steuerung	Stellt Optionen zum Finden von Pfaden für das Objekt zur Verfügung.
Sichtbarkeit	Ändert die Sichtbarkeit des Objektes.

Kameras in der Game Engine

Die Kamera die in der Game Engine genutzt wird ist ähnlich zu der, die im Blender Render System verwendet wird, und wird auf ähnliche Weise erstellt, parameterisiert und manipuliert. However because of its use as a real-time device, the Game Engine camera has a number of additional features - it may be used as not only as a static camera, but also as a moving device with its default characteristics (ie. with its own programmed moves), or it may track another object in the game. Furthermore, any game object may be used as a camera; the view is taken from the object's origin point. Lastly, it may be given special capabilities such as Stereo vision, Dome visualisation etc. which have special relevance to game technology.

When you start the Game Engine, the initial camera view is taken from the latest 3D View. This may be either a selected camera object or the default camera (see below). Thus to start the game with a particular camera, you must select the camera and press 0 NumPad before starting the Game Engine.

 **Zum Vermeiden von Kamera Verzerrungen, vergrößern Sie die Ansicht stets bis das Kameraobjekt den gesamten Ansichtspunkt füllt**
 {{{2}}}

Standard-Kamera

The default camera view is taken from the latest 3D viewport view, at a distance equivalent to the viewer. This means that, if the normal 3D view is active, the scene does not change when the Game Engine is started.

Kamera Objekt

Das Kameraobjekt in der Gameengine ähnelt sehr der konventionellen Blender Kamera - unter [Camera](#) finden sich dazu mehr Details, eine Kamera einzurichten, zu manipulieren und auszuwählen. Die folgenden Abteilungen klären über die speziellen Fassetten auf, die in den Blender Game Engine Kameras verfügbar sind.

Parent Camera to Object

Die Kamera wird dem Objekt folgen. Wählen Sie zuerst die Kamera und dann das Objekt aus. Danach StrgP → Elternteil setzen.

Beachten Sie dass wenn Ihr Objekt über Rotationen verfügt, wird die Kamera auch über die Rotationen verfügen. Um dies zu vermeiden benutzen Sie "Parent to Vertex" (siehe unten).

Parent to Vertex

The easiest way to accomplish this is to select your object and ⇧ Tab to Edit mode. Now select the vertex and ⇧ Tab back to Object mode.

Als nächsten, wählen Sie ohne irgendein ausgewähltes Objekt, die Kamera aus, und halten die ⇧ Shift Taste gedrückt, während Sie das Objekt auswählen. Wechseln Sie durch ⇧ Tab in den Editiermodus, dann StrlP und wählen Sie Vertex Elternteil setzen.

Jetzt wird die Kamera dem Objekt folgen und seine Rotation behalten, während das Objekt sich rotiert.

Objekt als eine Kamera

Jedes Objekt kann eine Kamera werden, mit was auch immer für welchen für das Objekt gesetzten Eigenschaften.

Um ein Objekt in eine Kamera zu versetzen, wählen Sie dazu im Objektmodus das Objekt aus, und drücken Sie auf dem NumPad Ctrl0 NumPad.

Um diese Sicht zurückzusetzen, wählen Sie einfach die Kamera mit Ctrl0 NumPad erneut aus.

Kameralinsen Verschiebung

In the blender interface, there is an option to shift the camera view on the x/y plane of the view. It is comparable to lens shift in video projectors, that usually shift the image up along the Y axis (so for ex. when you put the beamer on a table, it does not project half of the image on the table.)

Unfortunately, this parameter is not taken in account by the game engine.

To manipulate the projection, we can then modify directly the camera projection matrix in python.

```
import bge
scene = bge.logic.getCurrentScene()
cam = scene.active_camera
# get projection matrix
camatrix = cam.projection_matrix
#modifying the camera projection matrix by modifying the x and y terms of the 3rd row to obtain a shift of the rendered area
camatrix[2][0] = 2*shiftx
camatrix[2][1] = 2*shifty
cam.projection_matrix = camatrix
```

shiftx and shifty are here in field of view unit, so for ex. for shifting the view up half a screen, shifty is set to 0.5.

Note that a camera's projection_matrix attribute may not be set until after initialization scripts are executed and running this code

immediately after the game starts will mess up the projection matrix.

Blender Physik

Blender enthält erweiterte Physiksimulationen, in Form von der Bullet Physik Engine ([BulletPhysics.org](http://bulletphysics.org)). Most of your work will involve setting the right properties on the objects in your scene—dann können Sie sich zurück lehnen und die Engine übernimmt alles weitere. Die Physik Simulation kann für [Spiele](#), aber auch für Aufnahmen von [Animationen](#) genutzt werden.

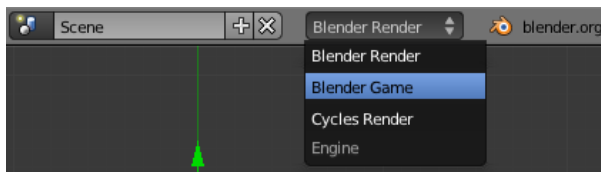
Die Blender Game Engine (BGE) basiert auf der Physik von Rigid-Body, welches sich signifikant von dem komplementären Set an Werkzeugen die in [Weichkörper Physik Simulationen](#) verwendet werden unterscheidet. Auch wenn die BGE über den Typ Weichkörper verfügt, ist es nicht annähernd so nuanciert wie nicht-BGE Weichkörper. Das Gegenteil ist mehr der Fall: es ist mit der kein-BGE Physik schwierig etwas das ähnlich zu einer steifen Form ist hinzubekommen. In Rigid Körper Physik ist, als Effekt oder Folge, keine Art von Mesh Deformationen vorhanden. Zu einer Diskussion wie dieser Effekt teilweise überwunden werden kann, können Sie sich [Mesh Deformationen](#) ansehen.

Erste Schritte

Wenn Sie noch nie mit der BGE gearbeitet haben, möchten Sie sich vielleicht 10-15 Minuten das [Einleitung Tutorial auf Englisch](#) ansehen. Anschließend folgt ein interaktives Beispiel, [Doc:2.6/Tutorials/Frijoles[Frijoles auf Englisch]].

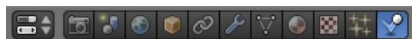
Die Grundlagen sind:

- Drücken Sie P während sich Ihre Maus in einer 3D Ansicht befindet um zu starten.
- Drücken Sie Esc zum Beenden.



Zur Blender Game Engine wechseln.

- Wechseln Sie zur "Blender Game" Engine damit Ihr Eigenschaften Editor die richtigen Optionen enthält.



Eigenschaften Knöpfe

- Beobachten Sie die Unterschiede zwischen "Blender Game" und "Blender Render" nachdem Sie verschiedene Eigenschaften Knöpfe gedrückt haben: Render, Szene, Welt, Materialien, Partikel, und, am wichtigsten, Physik.
- Beginnen Sie Änderungen an den Eigenschaften vorzunehmen um Ihre Simulationen zu beeinflussen.
- Take a step deeper with [Logic Bricks](#) for total control, to include Python scripting.

Typen

Die fünf Allzweck Typen sind:

Typ	Kollision	Bewegung	Rotation	Deformation	Beispiel
Keine Kollision					Eine Anzeige auf dem Bildschirm.
Statisch	✓				Eine Mauer.
Dynamisch	✓	✓			Ein Character in einem Scroller von der Seite.
Rigid Body	✓	✓	✓		Häufig bewegte Objekte.
Weicher Körper	✓	✓	✓	✓	Eine wehende Flagge.

Zusätzliche, spezielle Typen:

- [Occluder](#) - Verhindert Berechnungen von gerenderten Objekten (aber nicht deren Physik!).
- [Sensor](#) - Erkennt Objekte ohne die Restitution von Kollisionen.
- [Navigation Mesh](#) - Zum Erstellen von Pfaden zur Wegfindung. Nützlich für Künstliche Intelligenz.

Welt Optionen

The global Physics Engine settings can be found in the [World Properties](#), which include the Gravity constant and some important engine performance tweaks.

Beschränkungen

It is imperative to understand that the Blender Constraints generally don't work inside the BGE. This means interesting effects such as Copy Rotation are unavailable directly.

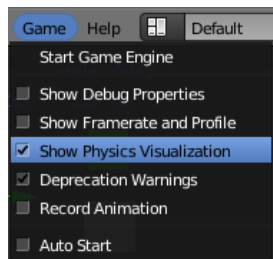
Your options include:

- [Parenting](#) - But not Vertex Parenting.
- [Rigid Body Joint](#) - This is the one Constraint that you can set up through the UI that works in the BGE. It has several options, and can be very powerful - see ITS page for a detailed description and demo .blend. Don't forget that you can loop through objects using `bpy` instead of clicking thousands of times to set up chains of these Constraints.
- Rigid Body Joints on the Fly - You can add/remove them after the BGE starts by using `bge.constraints.createConstraint()`.

This can be good either to simply automate their setup, or to truly make them dynamic. A simple demo can be viewed in: [BGE-Physics-DynamicallyCreateConstraint.blend](#)

- [Python Controllers](#) - As always, in the BGE, you can get the most power when you drop into Python and start toying with the settings directly. For instance, the Copy Rotation mentioned above is not hard -- All you have to do is something to the effect of
`own.worldOrientation = bge.logic.getCurrentScene().objects['TheTargetObject'].worldOrientation`

Physik Visualisieren



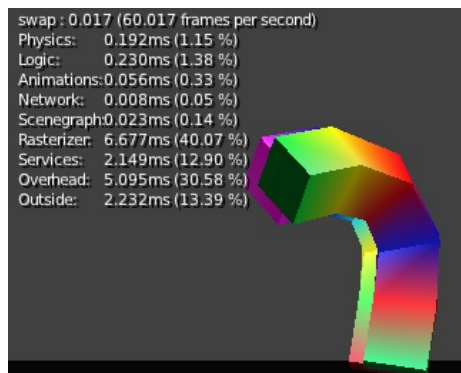
Go to Game » Show Physics Visualization to show lines representing various attributes of the Bullet representation of your objects. Note that these might be easier to see when you turn on Wireframe Mode (Z) before you press P. Also note that you can see how the Bullet triangulation is working (it busts all your Quads to Tris at run-time, but the BGE meshes are still quads at run-time).

- **RGB/XYZ Widget** - Representing the object's Local Orientation and Origin.
- **Green** - "sleeping meshes" that are not moving, saving calculations until an external event "wakes" it.
- **White** - White lines represent active bounding meshes that are undergoing physics calculations, until such calculations are so small that the object is put to rest. This is how you can see the effects of the Collision Bounds.
 - **Thick, or Many White Lines** - A compound collision mesh/meshes.
- **Violet** - Bounding meshes for Soft bodies.
- **Red** - The Bounding Box, the outer boundary of object. It is always aligned with global X Y and Z, and is used to optimize calculations. Also represents meshes that have been forced into "no sleep" status.
- **Yellow** - Normals.
- **Black** - When in wireframe, this is your mesh's visual appearance.

If you want finer-grained control over the display options, you can add this as a Python Controller and uncomment whichever pieces you want to see:

```
import bge
debugs = (
#bge.constraints.DBG_DRAWWIREFRAME, # Draw wireframe in debug.
bge.constraints.DBG_DRAWAABB, # Draw Axis Aligned Bounding Box in debug.
#bge.constraints.DBG_DRAWFEATURETEXT, # Draw features text in debug.
#bge.constraints.DBG_DRAWCONTACTPOINTS, # Draw contact points in debug.
#bge.constraints.DBG_NOHELPTXT, # Debug without help text.
#bge.constraints.DBG_DRAWTEXT, # Draw text in debug.
#bge.constraints.DBG_PROFILETIMINGS, # Draw profile timings in debug.
#bge.constraints.DBG_ENABLESATCOMPARISON, # Enable sat comparison in debug.
#bge.constraints.DBG_DISABLEBULLETLCP, # Disable Bullet LCP.
#bge.constraints.DBG_ENABLECCD, # Enable Continuous Collision Detection in debug.
#bge.constraints.DBG_DRAWCONSTRAINTS, # Draw constraints in debug.
#bge.constraints.DBG_DRAWCONSTRAINTLIMITS, # Draw constraint limits in debug.
#bge.constraints.DBG_FASTWIREFRAME, # Draw a fast wireframe in debug.
#bge.constraints.POINTTOPOINT_CONSTRAINT,
#bge.constraints.LINEHINGE_CONSTRAINT,
#bge.constraints.ANGULAR_CONSTRAINT,
#bge.constraints.CONETWIST_CONSTRAINT,
#bge.constraints.VEHICLE_CONSTRAINT,
)
for d in debugs:
    bge.constraints.setDebugMode(d)
```

Show Framerate and Profile



A shot of [Manual-BGE-Physics-DancingSticks.blend](#) with Game » Show Framerate and Profile enabled

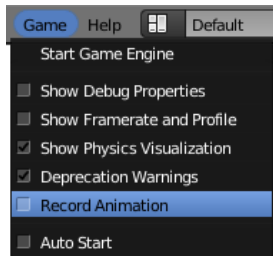
If you enable Game » Show Framerate and Profile, it will put some statistics in the upper-left area of the game window.

These can be very informative, but also a bit cryptic. Moguri has elaborated on their meanings, for us:

<http://mogurijin.wordpress.com/2012/01/03/bge-profile-stats-and-what-they-mean/>

Nach Schlüsselbildern aufnehmen

Abgesehen vom Spielen, möchten Sie manchmal eine komplexe Szene Rendern die Kollisionen, mehrere Kräfte, Friktion zwischen mehreren Körpern, und Air Drag beinhaltet---aber Sie haben nicht vor all diese einzeln zu animieren. Zum Glück, können Sie auf die Blender Game Engine zählen damit sie es für Sie erledigt.

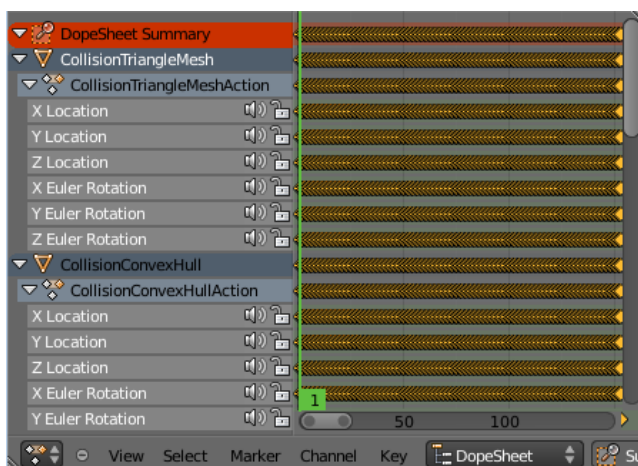


Menü zum Aufnehmen von Schlüsselbildern in das Dopesheet.

All you have to do to achieve this effect is go to the Info Editor (the bar at the top of the window) Game » Record Animation, and it will lock away your keyframes for use in Blender Render mode. You can go back to the 3D view and hit AltA to play it back, or CtrlF12 to render it out as an animation.

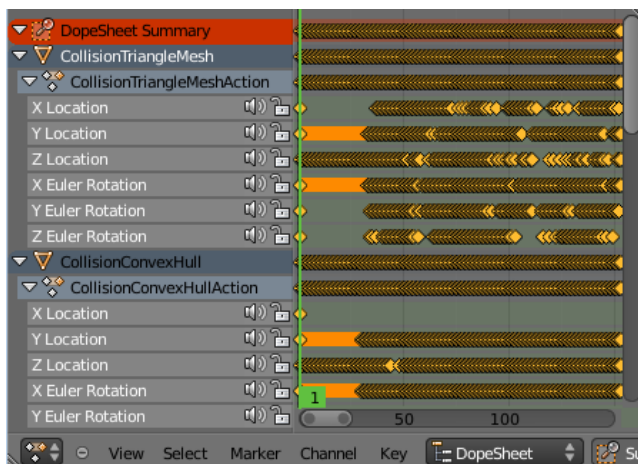
Note that, through use of Game Logic Bricks, you can interact with the Game Engine as it is playing. That means you can record a part-simulated part-user-controlled animation.

Schlüsselbilder aufräumen



The mess that "Record Animation" leaves behind.

You will find that Record Animation machine-guns keyframes all over the place. It will probably be a keyframe per channel per frame, wall-to-wall. Some of this data is redundant due to unchanging channels.



After hitting O.

Simply press O while in the DopeSheet and it will remove all keyframes that do not represent a value change compared to the one to

its left. Though this will not suddenly make your data dead simple, it will at least give you some clues about where the action/inaction areas are in the timeline.

Don't forget that the animation does not have to play starting from Frame 1. The keyframes will insert beginning at your current Frame position, or else you can use the [NLA Editor](#) and turn these things into Action Strips.

In .bullet Datei aufnehmen

You can snapshot the physics world at any time with the following code:

```
import bge
bge.constraints.exportBulletFile("test.bullet")
```

This will allow importing into other Bullet-based projects. See the [Bullet Wiki on Serialization](#) for more.

Mesh Deformationen

As mentioned above, Rigid Body physics do not affect mesh deformations, nor do they account for them in the physics model. This leaves you with a few options:

weichkörper

You can try using a [Soft Body](#), but these are fairly hard to configure well.

Aktionen

To use an [Action Actuator](#) to do the deformation, you have to make a choice. If you use Shapekeys in the Action, you will be fine as far as the overall collisions (but see below for the note on `reinstancePhysicsMesh()`). The mesh itself is both a display and a physics mesh, so there is not much to configure.

To use an Armature as the deformer will require a bit of extra thought and effort. Basically the Armature will only deform a mesh if the Armature is the parent of that mesh. But at that point, your mesh will lose its physics responsiveness, and only hang in the air (it's copying the location/rotation of the Armature). To somewhat fix this you can then parent the Armature to a collision mesh (perhaps a simple box or otherwise very-low-poly mesh). This "Deformation Mesh" will be the physics representative, being type: Dynamic or Rigid Body, but it will be set to Invisible. Then "Display Mesh" will be the opposite set to type: No Collision, but visible. This still leaves us with the problem mentioned in the previous paragraph.

When you deform a display mesh, it does not update the corresponding physics mesh. You can view this evidently when you [Enable Physics Visualization](#) - the collision bounds will remain exactly as when they began. To fix this, you must call `own.reinstancePhysicsMesh()` in some form. Currently this only works on Triangle Mesh bounds, not Convex Hull. We have prepared a demonstration file in [Manual-BGE-Physics-DancingSticks.blend](#). Note that we had to increase the World » Physics » Physics Steps » Substeps to make the collisions work well. The more basic case is the case the Shapekeyed Action, which you can see in the back area of the scene. Since it is the only object involved, you can call `reinstancePhysicsMesh()` unadorned, and it will do the right thing.

The more complicated case is the Collision Mesh » Armature » Display Mesh cluster, which you can see in the front of the scene. What it does in the .blend is call `reinstancePhysicsMesh(viz)`, that is, passing in a reference to the visual mesh. If we tried to establish this relationship without the use of Python, we would find that Blender's dependency check system would reject it as a cyclic setup. This is an example of where Blender's checking is too coarsely-grained, as this circle is perfectly valid: the grandparent object (the Collision Mesh) controls the location/rotation, while the middle object (the Armature) receives the animated Action, where the child (the Display Mesh) receives the deformation, and passes that on up to the top, harmlessly. Something to note is that the Collision Mesh is merely a plane -- that is all it requires for this, since it will be getting the mesh data from `viz`.

Ragdolls

A third option is to create your items out of many sub-objects, connected together with Rigid Body Joints or similar. This can be quite a bit more work, but the results can be much more like a realistic response to collisions. For an Addon that can help you out in the process, check out the [Blender Ragdoll Implementation Kit](#).

Tieferes Wissen

Sometimes you will want to look at:

- The main Bullet Physics page - <http://bulletphysics.org/wordpress/>
- The Bullet Wiki - <http://www.bulletphysics.org/mediawiki-1.5.8/index.php?title=Documentation>
- The Bullet API Docs - <http://www.continuousphysics.com/Bullet/BulletFull/index.html>
- The Bullet Forums - <http://www.bulletphysics.org/Bullet/phpBB3/>

Und dann existiert immer die Option:

Blender und Bullet Quellcode Dateien lesen

Das könnte einschüchtern klingen, sogar wenn Sie C/C++ kennen, aber es kann sehr informativ sein. Sie können sehen wie Blender Objekte einrichtet um Sie Bullet zu übergeben, und an einigen Stellen `printf()` Anweisen an hinzufügen, oder mit `svn revert`

experimentieren um zur Normalität zurückzukehren.

Here is an example of the trail to get to the bottom of the handling of the options. We will observe the handling of the `use_shape_match` property, as an example.

- Start by getting [The Blender Source Tree](#)
- If you search it for `use_shape_match` (e.g., by `grep -r use_shape_match .`), this will lead you to [blender/source/blender/makesrna/intern/rna_object_force.c](#), which says:

```
prop = RNA_def_property(srna, "use_shape_match", PROP_BOOLEAN, PROP_NONE);
RNA_def_property_boolean_sdna(prop, NULL, "flag", OB_BSB_SHAPE_MATCHING);
RNA_def_property_ui_text(prop, "Shape Match", "Enable soft body shape matching goal");
```

- From this we see that the internal flag is set from the value of `OB_BSB_SHAPE_MATCHING`
- Searching for that leads us to:
 - Its simple initialization in [blender/blenkernel/intern/bullet.c](#)
 - Its assignment to `objprop.m_gamesoftFlag`, an object of type `KX_ObjectProperties`, in [gameengine/Converter/BL_BlenderDataConversion.cpp](#) -- so far, only passing the value, no actual decision-making.
- Searching for that leads us to [gameengine/Physics/Bullet/CcdPhysicsController.cpp](#) where we can find the following:

```
if (m_cci.m_gamesoftFlag & CCD_BSB_SHAPE_MATCHING) //OB_SB_GOAL)
{
    psb->setPose(false,true); //
} else
{
    psb->setPose(true,false);
}
```

- Here is the first bit of logic. It inverts the arguments to `setPose` depending on the value. Now then, since `psb` is of type `btSoftBody`, we have officially launched into Bullet territory. You have a couple options:
 - If you go to the [Bullet API Navigator](#) and expand the Class List menu, you can CtrlF for the `btSoftBody` class, and follow the link to the [btSoftBody Class Reference](#) Page. There you will see very sparse written documentation, but it will, at least, link you to a syntax-highlighted [line](#) where the method is implemented.
 - Get the Bullet Source with: `svn checkout http://bullet.googlecode.com/svn/trunk/ bullet-read-only` and probably run something like `ctags -r .` from that tree every now and then to build the `tags` file. Now you can dig further. Something like `vim -t setPose` will lead you to the implementation in [src/BulletSoftBody/btSoftBody.cpp](#) (which is the same code as can be found through the Bullet API Navigator in the previous step).
- Through either approach, we find that the mysterious `bools` above are for `btSoftBody::setPose(bool bvolume,bool bframe)`, which are immediately assigned to `m_pose.m_bvolume` and `m_pose.m_bframe`, respectively.
 - Subsequently searching for `m_bvolume` doesn't show much use in this file, other than the assignment and initialization. We could follow the trail deeper to the [btSoftBody::Pose Struct Reference](#) docs, but for now let's try:
 - Searching for `m_pose.m_bframe`. At this point, in this file, we have finally found the end of the simple passing of the flags, and we will see major chunks of code that are branched depending on this setting.
- Whether we can learn anything apparent at this point will depend on our ability to understand the code and concepts within the Bullet implementation. Perhaps we followed a multi-step process to find inscrutability, but at least we can see the very lines executed within the BGE.
 - Now we have some symbols to search for in [Google](#) or in the [Bullet Forums](#).
 - If we wanted to instrument this code with debugging `printf()`s, we could compile it and link it into our Blender build.

Eigenschaften

Eine umfassende Liste der verfügbaren Funktionen in der Blender Game Engine:

Editor

- Plattformen: GNU/Linux, Windows, Mac
- Logic brick Arbeitsablauf.
- Python Scripting.
- Schattenkarten Backen
- Lichtkarten Backen
- Eingebauter Player (WYSIWYG)
- Profiling während der Laufzeit
- Physik Visualisierung (debugging)

Engine

- Mehrere Shading Modi
 - GLSL Shading
 - Multitexturen
 - Einzeltexturen
- GLSL Shader (2.0, Vertex und Fragmente - Python oder fest eingebaut).
 - Normale Karte
 - Parallax Karte
 - Glanzlicht Karte
 - Farb Karte
 - Detail Karte
 - Umgebungs Reflexion (Würfel,Kugel)
 - Vertex Farbe
 - Zimmer Shader
 - Dynamische Schatten (Soft shadows and Stencil shadows; harmony branch)
- Physik: Bullet
 - Auswahl System: Kugel, Box, Stumpf (durch Python)
 - Kollision System: capsule, box, sphere, cylinder, cone, convex hull, triangle mesh
- Klang: WAV, MP3, Ogg
- Texturen: tga(+Alphakanal), png(+Alphakanal), jpg, Videotexturen Support
- Text: Bitmap Schriftarten und Schriftarten Objekte
- Pfad Findung: Navigation Mesh mit mit Hindernisvermeidung; Neufassung und Detour
- Animation: Mesh Deformation mit Knochen (Skelette)
- Mip-mapping
- Anti-Aliasing (MSAA)

Unvollständig / Zukünftige Eigenschaften

- GL Partikel System (Siehe xEmitter)
- Terrain Engine
- GUI (siehe BGUI)
- LOD (Level an Details - durch Python).
- Portale (noch nicht).
- Netzwerk (durch Python).
- Multi UV Koordinaten (durch Python).